

# **Intelligentes Führungskonzept für ein Autonomes Unterwasserfahrzeug in Sondersituationen**

**Dissertation**  
**zur Erlangung des akademischen Grades**  
**Doktoringenieur (Dr.-Ing.)**

**vorgelegt der Fakultät für Informatik und Automatisierung  
der Technischen Universität Ilmenau**

**von Dipl.-Ing. Mike-Joachim Eichhorn**  
**geboren am 14. Mai 1966 in Eisfeld**

**Gutachter: Prof. Dr.-Ing. habil. J. Wernstedt, Fraunhofer-AST Ilmenau**  
**Dr.-Ing. T. Kuhn, Diehl BGT Defence GmbH Überlingen**  
**Dr. R. Bachmayer, National Research Council Canada St. John's**

**Tag der Einreichung: 11. September 2006**

**Tag der wissenschaftlichen Aussprache: 20. April 2007**



# Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter im Rahmen des BMBF-Forschungsprojektes „DeepC“ am Fachgebiet Systemanalyse des Instituts für Automatisierungs- und Systemtechnik der Technischen Universität Ilmenau.

Ich möchte mich bei Herrn Prof. Dr.-Ing. J. Wernstedt bedanken, welcher mir die Möglichkeit gab, dieses interessante Forschungsthema zu bearbeiten, bei Dipl.-Ing. Marco Jacobi für die softwaretechnische Unterstützung und den regen Erfahrungsaustausch bei der Entwicklung der Graphenmethoden sowie bei meinem Kollegen Dr.-Ing. Divas Karimanzira für die gute Zusammenarbeit und Hilfe während des Projektes.

Herrn Dr. R. Bachmayer (National Research Council Canada, Institute for Ocean Technology, St. John's) danke ich für die hilfreichen Diskussionen und die Übernahme eines Koreferats. Für die Übernahme des weiteren Koreferats danke ich Herrn Dr.-Ing. T. Kuhn (Diehl BGT Defence GmbH, Überlingen).

Ich danke den Herren Dipl.-Ing. Uwe Möller, Dipl.-Ing. Jörg Kalwa, Dipl.-Ing. Hendrik Schelenz und Dipl.-Ing. Erik Miszewski von ATLAS ELEKTRONIK GmbH Bremen für die gute Kooperation während der dreijährigen Projektlaufzeit.

Mr. Sandy McPherson CEng MBCS von Allset Consultancy BV danke ich für seine kollegiale und bereichernde Zusammenarbeit bei der Softwareentwicklung.

Für die fruchtbaren Diskussionen bei der Durchsicht des Manuskriptes danke ich Herrn Dr.-Ing. Hartmut Linke. Meinem Arbeitskollegen und Tauchlehrer Dipl.-Ing. Wolfgang Schulke möchte ich für die mir zuteil gewordene Unterstützung danken.

Mein besonderer Dank gilt Frau Dr. rer. nat. Uta Kirschnick, welche durch ihre sorgfältige und gewissenhafte Durchsicht des Manuskriptes wesentlich zum Gelingen dieses Buches beitrug.

Den Korrekturlesern Dipl.-Sozialpäd. Monika Dittrich, Christiane und Senta May, Dipl.-Ing. Jörg Eichhorn, Dipl.-Ing. Marco Jacobi und Dr.-Ing. Volker Zerbe danke ich für ihre tatkräftige Unterstützung.

Meinen Eltern möchte ich für die persönliche Unterstützung danken.

*„Wenn Du ein Schiff bauen willst, dann trommle nicht Männer zusammen,  
um Holz zu beschaffen, Aufgaben zu vergeben und die Arbeit einzuteilen,  
sondern lehre die Männer die Sehnsucht nach dem weiten endlosen Meer.“*

Antoine de-Saint Exupéry, französischer Schriftsteller, 1900-1944

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung.....</b>	<b>1</b>
1.1	<i>Motivation und Einordnung der Arbeit.....</i>	2
1.1.1	Forschungsprojekt „DeepC“ .....	2
1.1.2	Versuchsträger AUV M63 .....	3
1.1.3	Einsatzmöglichkeiten für AUVs .....	4
1.1.4	Hierarchisches Führungskonzept des Fahrzeuges.....	6
1.1.5	Aufbau und Aufgaben des Moduls FIS.....	8
1.2	<i>Anforderungen, Vorgaben und Bedingungen.....</i>	10
1.2.1	Anforderungen .....	10
1.2.2	Hardwaretechnische Vorgaben .....	11
1.2.3	Softwaretechnische Vorgaben.....	13
1.2.4	Arbeitsbedingungen .....	14
1.3	<i>Beschreibung der Datenstrukturen .....</i>	15
1.3.1	Objektmodellierung.....	15
1.3.2	Bereichsdefinitionen.....	17
1.3.3	Missionsplan.....	17
1.4	<i>Zielsetzung, Aufbau und Beitrag der Arbeit.....</i>	19
<b>2</b>	<b>Kollisionsüberwachung.....</b>	<b>21</b>
2.1	<i>Einführung in die Kollisionserkennung.....</i>	21
2.1.1	Bekannte Ansätze .....	22
2.1.2	Einsatzfälle bei autonomen mobilen Systemen und in der Robotik.....	23
2.2	<i>Konzept der Kollisionsüberwachung .....</i>	24
2.2.1	Anforderungen .....	24
2.2.2	Vorstellung des Verfahrens .....	24
2.3	<i>Kollisionserkennung.....</i>	25
2.3.1	Problemstellung.....	25
2.3.2	Bestimmung der Kollisionsparameter .....	26
2.3.2.1	Strecke – Elliptischer Zylinder .....	26
2.3.2.2	Kreisbogen – Elliptischer Zylinder .....	28
2.3.3	Das Zwei-Phasen-Konzept zur Kollisionserkennung .....	29
2.3.4	Erste Stufe der Weiten Phase .....	31
2.3.4.1	Überlappungstest zwischen achsengerichteten Boxen.....	32
2.3.5	Zweite Stufe der Weiten Phase .....	33
2.3.5.1	Überlappungstest zwischen orientierten Boxen .....	34

2.3.5.2	Überschneidungstest zwischen einer Strecke und einer orientierten Box....	36
2.4	<i>Laufzeitmessungen</i> .....	38
2.4.1	Laufzeiten zur Erzeugung der einzelnen Attribute .....	38
2.4.2	Laufzeiten der einzelnen Überlappungs- bzw. Schnitttests .....	39
2.4.3	Bestimmung der Schnittwahrscheinlichkeiten .....	40
2.4.4	Vergleich unterschiedlicher Konzepte .....	41
2.5	<i>Zusammenfassung</i> .....	44
<b>3</b>	<b>Kollisionsvermeidungssystem</b> .....	<b>45</b>
3.1	<i>Einführung in die Kollisionsvermeidung</i> .....	45
3.1.1	Begriffsbestimmung .....	45
3.1.2	Vergleich von planendem und reaktivem Ansatz .....	47
3.1.3	Bekannte Realisierungen.....	49
3.2	<i>Anforderungen an die Kollisionsvermeidung</i> .....	51
3.2.1	Objektdefinition .....	51
3.2.2	Operationsgebiet.....	51
3.2.3	Manövrierfähigkeit und Umweltbedingungen .....	52
3.2.4	Ausweichen in der vertikalen Ebene .....	52
3.2.5	Wegroute .....	54
3.2.6	Anforderungskatalog.....	54
3.3	<i>Architektur des Kollisionsvermeidungssystems</i> .....	54
3.4	<i>Zusammenfassung</i> .....	56
<b>4</b>	<b>Wegeplanung</b> .....	<b>57</b>
4.1	<i>Einführung in die Wegeplanung</i> .....	57
4.1.1	Bekannte Ansätze .....	57
4.1.2	Darstellungsmöglichkeiten der Umwelt in einem Graphen .....	59
4.1.2.1	Gitterbasierte Darstellung .....	59
4.1.2.2	Quadtree- / Octreegraph .....	59
4.1.2.3	Sichtbarkeitsgraph.....	60
4.1.2.4	Kreistangentenansatz.....	61
4.1.3	Suchalgorithmen für kürzeste Wege in einem Graphen.....	61
4.2	<i>Beschreibung der Wegeplanung</i> .....	62
4.2.1	Anforderungen .....	62
4.2.2	Vorstellung des Konzeptes.....	62
4.3	<i>Grapherzeugung</i> .....	64
4.3.1	Quadtree- / Octreegraph .....	64
4.3.1.1	Zerteilungsverfahren des Operationsgebietes .....	64
4.3.1.2	Ermittlung der Nachbarschaftsbeziehungen.....	67
4.3.2	Sichtbarkeitsgraph.....	70
4.3.2.1	Bestimmung der Knoten.....	70
4.3.2.2	Ermittlung der Kanten .....	71

4.3.2.3	Sektorisierung des Operationsgebietes.....	71
4.3.3	Bestimmung des Kostenwertes .....	74
4.4	<i>Erzeugung des Routenplanes</i> .....	76
4.4.1	Routenplanung mit der Boost Graph Library.....	76
4.4.1.1	Datenstrukturen .....	76
4.4.1.2	Suchverfahren.....	76
4.4.2	Verfahren zur Glättung des Weges .....	77
4.5	<i>Erzeugung des Manöverplanes</i> .....	79
4.5.1	Erzeugung des Manöverplanes.....	79
4.5.2	Überprüfung des Manöverplanes .....	82
4.6	<i>Abarbeitung des Manöverplanes</i> .....	83
4.6.1	Bestimmung der Bahnparameter für die Strecke .....	83
4.6.2	Bestimmung der Bahnparameter für den Kreisbogen .....	84
4.7	<i>Test der Verfahren</i> .....	85
4.7.1	Hindernisparcours .....	85
4.7.2	Praktisches Testszenario .....	89
4.8	<i>Zusammenfassung</i> .....	92
<b>5</b>	<b>Reaktive Steuerung</b> .....	<b>93</b>
5.1	<i>Aufgabenbereich und Anforderungen der Reaktiven Steuerung</i> .....	93
5.2	<i>Übersicht über bekannte Realisierungen von reaktiven Ansätzen</i> .....	94
5.2.1	Künstliche Potentialfelder .....	94
5.2.2	Fuzzybasierte Steuerung .....	96
5.2.3	Vektorfeld-Histogramm .....	98
5.2.4	Harmonische Dipol-Potentiale .....	99
5.3	<i>Verfahren der Geometrischen Konstruktion</i> .....	101
5.3.1	Erzeugung von Gradientenlinien durch geometrische Konstruktion .....	101
5.3.2	Einbeziehung der Geschwindigkeitsinformationen .....	103
5.3.3	Anwendung bei elliptischen Sicherheitszonen.....	104
5.3.4	Anwendung für mehrere Hindernisse .....	106
5.3.5	Überlappende Sicherheitsbereiche .....	109
5.3.5.1	Bildung der Hüllsicherheitszonen .....	109
5.3.5.2	Schnittberechnung Ellipse – Ellipse.....	110
5.3.5.3	Erzeugung der Ellipse einer Hüllsicherheitszone.....	111
5.3.5.4	Bildung des Gradienten im Inneren der Hüllsicherheitszone.....	113
5.4	<i>Test des Verfahrens</i> .....	113
5.4.1	Vergleich der Verfahren.....	113
5.4.2	Führung des Fahrzeuges.....	114
5.4.3	Laufzeitmessungen.....	115
5.5	<i>Zusammenfassung</i> .....	118
<b>6</b>	<b>Identifikationssystem</b> .....	<b>119</b>

6.1	<i>Anwendungen der Objektidentifikation</i> .....	119
6.2	<i>Konzept der Objektidentifikation</i> .....	121
6.2.1	Anforderungen und Aufgaben.....	121
6.2.2	Vorstellung des entwickelten Konzeptes .....	121
6.3	<i>Ausrichten auf ein Objekt</i> .....	122
6.3.1	Ablauf der Identifikation.....	122
6.3.2	Bestimmung des nächstgelegenen Punktes auf der Ellipse.....	124
6.3.3	Test des Verfahrens .....	125
6.4	<i>Umfahren eines Objektes</i> .....	126
6.4.1	Ablauf der Identifikation.....	126
6.4.2	Test des Verfahrens .....	128
6.5	<i>Zusammenfassung</i> .....	128
<b>7</b>	<b><i>Zielpunktgenerierung</i> .....</b>	<b>129</b>
7.1	<i>Anwendungen zur Zielpunktgenerierung</i> .....	129
7.2	<i>Konzept der Zielpunktgenerierung</i> .....	131
7.2.1	Anforderungen .....	131
7.2.2	Vorstellung des Verfahrens .....	131
7.3	<i>Bemerkungen zum Laufzeitverhalten</i> .....	133
7.4	<i>Zusammenfassung</i> .....	134
<b>8</b>	<b><i>Führungsalgorithmen</i> .....</b>	<b>135</b>
8.1	<i>Bekannte Ansätze</i> .....	135
8.2	<i>Regelungstechnische Analyse</i> .....	137
8.2.1	Regelungshierarchie .....	137
8.2.2	Modellierung des Fahrzeuges.....	138
8.3	<i>Synthese der Führungsregelungen</i> .....	140
8.3.1	Anforderungen an die Regelungen.....	140
8.3.2	Adaptiver PID-Regler .....	141
8.3.3	Bahnregelung .....	142
8.3.4	Kursregelung .....	142
8.3.5	Abstandsregelung .....	143
8.3.6	Peilungsregelung .....	143
8.4	<i>Nichtlinearer Regelungsentwurf</i> .....	144
8.4.1	Reglerparameterbestimmung mittels Parameteroptimierung.....	144
8.4.2	Gütekriterien im Zeitbereich .....	145
8.4.3	Robustheit.....	146
8.5	<i>Robustheitsentwurf der Kursregelung</i> .....	147
8.6	<i>Zusammenfassung</i> .....	148



<b>9</b>	<b>Test und Erprobung.....</b>	<b>149</b>
9.1	<i>Simulations- und Testumgebung unter MATLAB®/Simulink® .....</i>	<i>149</i>
9.2	<i>Simulative Untersuchung der Kollisionsvermeidung.....</i>	<i>150</i>
9.3	<i>Seeversuche .....</i>	<i>153</i>
9.4	<i>Zusammenfassung .....</i>	<i>154</i>
<b>10</b>	<b>Zusammenfassung und Ausblick .....</b>	<b>155</b>
10.1	<i>Zusammenfassung .....</i>	<i>155</i>
10.2	<i>Ausblick.....</i>	<i>157</i>
	<b>Anhang .....</b>	<b>158</b>
A.1	<i>Mathematische Beziehungen.....</i>	<i>158</i>
A.1.1	<i>Quadratische Gleichung der Ellipse.....</i>	<i>158</i>
A.1.2	<i>Lösung eines Systems aus zwei Variablen und zwei quadratischen Gleichungen .....</i>	<i>159</i>
A.2	<i>Flussdiagramme.....</i>	<i>161</i>
A.3	<i>Darstellung der geometrischen Graphen.....</i>	<i>163</i>
	<b>Literaturverzeichnis.....</b>	<b>166</b>

# Variablen und Abkürzungen

## Variablen

$a_{\text{Ellipse}}$	Hauptachse der Ellipse
$b_{\text{Ellipse}}$	Nebenachse der Ellipse
$\theta_{\text{Ellipse}}$	Rotationswinkel der Ellipse
$e_{\alpha G}$	Regelabweichung der Kursregelung
$e_{\rho}$	Peilungswinkelabweichung
$e_{\text{Ablage}}$	Bahnablage
$\mathbf{G}$	Gradient
$t_{\text{coll}}$	Kollisionszeit
$\rho$	Peilungswinkel
$P(S)$	Schnittwahrscheinlichkeit
$Q$	Gütefunktionswert
$\mathbf{v}_{\text{Strömung}}$	Strömungsvektor
$\mathbf{v}_{\text{Bahn}}$	Vektor der Bahnrichtung
$v_{\text{Bahn\_ef}}$	Erd feste Bahngeschwindigkeit
$\psi_{\text{Bahn}}$	Bahnrichtung
$w_{\alpha G}$	Sollwert der Kursregelung
$w_{\text{dist}}$	Sollabstand
$w_{\psi}$	Sollgierwinkel
$\mathbf{x}_{\text{Ellipse}}$	Mittelpunkt der Ellipse
$\mathbf{x}_{\text{coll}}$	Kollisionsposition
$\mathbf{x}_{\text{nah}}$	Nächstgelegene Position auf einem Objekt zu einer vorgegebenen Position
$\mathbf{x}_{\text{End}}$	Endposition des Manövers
$\mathbf{x}_{\text{Fahr}}$	Fahrzeugposition
$\mathbf{x}_{\text{Hind}}$	Hindernisposition
$\mathbf{x}_{\text{Start}}$	Startposition der Routenplanung oder des Manövers
$\mathbf{x}_{\text{Ziel}}$	Zielposition der Routenplanung oder anzufahrender Zielpunkt

## Abkürzungen

AABB	Axis-aligned Bounding Box / Achsengerichtete Begrenzungsbox
AUV	Autonomous Underwater Vehicle / Autonomes Unterwasserfahrzeug
BGL	Boost Graph Library
CFK	Kohlenfaserverstärkte Verbundstoffe
CGAL	Computational Geometry Algorithms Library
DVL	Doppler Velocity Log
FCD	Fuzzy Control Design
FdW	Fahrt durchs Wasser
FIS	Fahrzeugführung in Sondersituationen
FüG	Fahrt über Grund
GPS	Global Positioning System
INS	Inertiales Navigationssystem
NCD	Nonlinear Control Design
OBB	Oriented Bounding Box / Orientierte Begrenzungsbox
OG	Octreegraph
ROV	Remotely Operated Vehicle / Ferngesteuertes Unterwasserfahrzeug
SG	Sichtbarkeitsgraph
SSG	Sektorisierter Sichtbarkeitsgraph
QG	Quadtreegraph
WP	Wegpunkt

# Zusammenfassung

Die vorliegende Arbeit stellt ein vollständiges und neues Konzept zur Fahrzeugführung in Sondersituationen für ein Autonomes Unterwasserfahrzeug vor. Eine Sondersituation ist dann gegeben, wenn während einer Mission Objekte den abzufahrenden Routenplan tangieren. Die möglichen Handlungen bestehen im Ausweichen oder in der Identifikation dieser Objekte. Bei der Erstellung des Konzeptes gab es eine Reihe praxisrelevanter Anforderungen (Sicherheit, Robustheit, Rechenzeit, Optimalität) an die zu entwickelnden Verfahren und Algorithmen, die unter den hard- und softwaretechnischen Vorgaben und Arbeitsbedingungen einzuhalten waren. Solche Vorgaben umfassen das nichtholonome, verzögerte Bewegungsverhalten des Unterwasserfahrzeuges, die Sensorik zur Bestimmung der Umwelt, die im Fahrzeug eingesetzte Rechentechnik sowie die zu verwendende Entwicklungs- und Kommunikationssoftware. Die Arbeitsbedingungen eines Unterwasserfahrzeuges sind durch ein Manövrieren im dreidimensionalen Raum bei einer möglichen Seeströmung, schlechter Sicht und hohem Wasserdruck unter Einhaltung eines Sicherheitsabstandes zum Meeresbodens und zu den geographischen Hindernissen, technischen Bauten und Altlasten charakterisiert.

Das entwickelte Konzept ist modular aufgebaut und umfasst Komponenten zur Kollisionsüberwachung, Zielpunktgenerierung, Kollisionsvermeidung, Fahrzeugführung bei Identifikationsaufgaben sowie zur Fahrzeugsteuerung. Für die Kollisionsüberwachung wird ein Zwei-Phasen-Konzept eingesetzt. Dieses Konzept ermöglicht eine schnelle Kollisionsüberprüfung durch die Verwendung einfacher Kollisionstests zur Vorselektion möglicher Kollisionskandidaten. Das in dieser Arbeit entwickelte Kollisionsvermeidungssystem besitzt eine hybride Struktur, bei der ein reaktiver (Reaktive Steuerung) und ein planender Ansatz (Wegeplanung) parallel arbeiten. Die Reaktive Steuerung übernimmt die Führung des Fahrzeuges, während die Wegeplanung einen Routenplan generiert. Steht ein Routenplan zur Verfügung, arbeitet die Wegeplanung diesen ab.

Für die Reaktive Steuerung wurde ein neues Verfahren zur geometrischen Konstruktion von Gradientenlinien entwickelt. Es verbindet die Vorteile des von Guldner entwickelten Verfahrens der Harmonischen Dipolpotentiale mit der Forderung einer wegoptimalen Fahrweise unter Berücksichtigung des nichtholonomen, verzögerten Bewegungsverhaltens des Fahrzeuges. Durch die Forderung der online-Erzeugung eines Routenplanes unter Gewährleistung des Echtzeitverhaltens des Systems wurden graphenbasierte Verfahren für die Wegeplanung untersucht. Diese Verfahren ermöglichen es, einen optimalen Weg nach definierten Vorgaben in einer kalkulierbaren Zeit zu ermitteln. Zwei neu entwickelte Verfahren zur Generierung eines geometrischen Graphen aus einem Konfigurationsraum mit elliptischen Objektzylindern sowie ein Algorithmus zur Bestimmung der Fahrkosten unter Einbeziehung der Strömungsinformation werden detailliert beschrieben. Obgleich das in dieser Arbeit vorgestellte Führungskonzept für ein Autonomes Unterwasserfahrzeug entwickelt wurde, können Teile dieser Arbeit auch für boden- und luftgeführte Autonome Mobile Systeme angewandt werden.

# Abstract

This thesis introduces a complete and new concept for the control of an autonomous underwater vehicle (AUV) in special situations. Such a special situation occurs when an Object detected during the mission is tangential to the proposed route; in such a situation the possible actions are identification or evasion of the object. The design of the concept had to take into account a number of practical requirements (Safety, Robustness, Computation time and Optimality of the solution), for the processes and algorithms; these requirements had to be met within the hardware and software specifications and operational constraints of the AUV system. Such specifications include non-holonomic, delayed motion behaviour of the AUV, the available environmental sensors, the vehicle's software architecture, development and communication software. The operational constraints of a AUV may be characterised by manoeuvres in three dimensional space in adverse conditions with strong water currents, bad (sonar) visibility and high water pressure; taking into account safety distances to the seafloor and geographical obstacles, manmade constructions and debris.

The concept developed is of modular construction and includes components for collision detection, goal generation, collision avoidance, vehicle guidance for identification tasks and general vehicle control. A two phase concept was used for the collision detection; this allows a rapid collision verification through the use of simple collision tests which leads to a pre-selection of possible collision candidates. The collision avoidance system, developed during the course of the research, has a hybrid structure, whereby reactive (Reactive Control) and planning (Route planning) components work in parallel. The reactive control takes over the vehicle guidance to avoid the collision, concurrently the route planning generates a route which will avoid the obstacle and return the vehicle back on to its original path. Once a route has been calculated the planning function takes over from the reactive control to execute the planned route.

The new reactive control component contains a newly developed and original process for construction of gradient lines that combines the advantages of Harmonic Dipole Potentials process with the requirements of a path-optimal control which takes into account the non-holonomic, delayed motion behaviour of the vehicle. Due to the demands of the online generation of a route whilst guaranteeing the real-time behaviour of the control system as a whole, graph based techniques for route planning were investigated in the course of the research. These techniques allow an optimal path according to defined input to be calculated within a predictable time. Two newly developed techniques for geometrical graph generation from a configuration space with elliptic-cylindrical objects and an algorithm for calculating the energy requirements (inclusive of water-current data) are described in detail. Even though the guidance concept presented in this thesis was developed for an autonomous underwater vehicle, the concept or parts of the concept are equally applicable to land based or aerial mobile autonomous systems.



# 1 Einführung

Das Meer bedeckt über 71 % der Erdoberfläche. Seine Nutzung und Erforschung sind eng mit der Menschheitsgeschichte verbunden. Stand das Meer zuerst nur als Nahrungslieferant und Transportweg zur Verfügung, erfolgte mit der Industrialisierung Mitte des 19. Jahrhunderts die Verlegung von Telekommunikationskabeln auf seinem Boden. Mitte des letzten Jahrhunderts begann die industrielle Erschließung und Gewinnung seiner Öl- und Gasvorkommen. Mit der vermehrten Nutzung des Meeres nahm aber auch seine Verschmutzung in den letzten Jahrzehnten erheblich zu. So sind eine Überwachung der Ozeane sowie die Suche und Beseitigung von Schadstoffen wichtige Aufgaben im Umweltschutzbereich geworden.

In diesen maritimen Industrie- und Dienstleistungsbereichen nehmen unbemannte ferngesteuerte Unterwasserfahrzeuge (Remotely Operated Vehicles ROV) eine zentrale Rolle bei der Seebodenvermessung, Trassen- und Fernerkundung, Inspektion, Suche und Überwachung ein. Sie ermöglichen ein Arbeiten in für Taucher unzugänglichen Tiefen. Durch die hohe Dämpfung des Wassers ist eine Funk- oder Akustikverbindung zur Fahrzeugsteuerung nur bedingt einsetzbar, so dass die ROVs kabelgebunden arbeiten. Dies schränkt allerdings den Aktionsradius sowie die Einsatzgebiete eines solchen Fahrzeuges ein. Des Weiteren wirkt mit zunehmender Tiefe das Kabel wie ein zusätzlicher Strömungswiderstand und beschränkt so die Manövrierfähigkeit des Fahrzeuges. Die ständige Führung des Fahrzeuges von einem Trägerschiff aus erfordert entsprechende Wetterbedingungen und führt zu hohen Betriebskosten.

Aus diesen Gründen werden weltweit Bemühungen unternommen, um Aufgabenbereiche der ROVs von Autonomen Unterwasserfahrzeugen (Autonomous Underwater Vehicle AUV) durchführen zu lassen. Diese Fahrzeuge benötigen keinen Operator. Sie besitzen eine eigene autonome Mobilität und können als Autonome Mobile Systeme klassifiziert werden. Die Aufgaben des Operators bei einem ROV übernehmen bei einem AUV die im Fahrzeug befindlichen Rechneinheiten. Die Intelligenz zur Führung eines Fahrzeuges wird quasi vom Operator an die Rechneinheiten mit den Entscheidungs- und Steueralgorithmen übergeben.

Eine Teilaufgabe dieser Intelligenz ist die *Fahrzeugführung in Sondersituationen*. Die vorliegende Arbeit stellt ein vollständiges Konzept dafür vor. Unter Sondersituationen sind die Kollisionsvermeidung und die Identifikation von Objekten zu verstehen. Die hierbei zu lösenden Aufgaben sind von vielfältiger Natur. Da es keine Beschränkungen auf spezielle Fachgebiete oder Lösungsansätze bei der Erstellung des neuen Konzeptes gab, sind die zu entwickelnden Ansätze in den Bereichen der Mathematik, der Regelungstechnik und der Informatik angesiedelt. Diese interdisziplinäre Betrachtungsweise gestattet es, den vorgegebenen praxisrelevanten Anforderungen an das Konzept gerecht zu werden.

Obgleich das in dieser Arbeit vorgestellte Führungskonzept für ein Autonomes Unterwasserfahrzeug entwickelt wurde, können Teile dieser Arbeit auch für boden- oder luftgeführte Autonome Mobile Systeme angewandt werden.

# 1.1 Motivation und Einordnung der Arbeit

## 1.1.1 Forschungsprojekt „DeepC“

Die vorliegende Arbeit stellt ein Teilgebiet des Forschungsprojektes „DeepC“ [8] vor. Das Ziel dieses Projektes bestand in der Entwicklung eines Autonomen Unterwasserfahrzeuges für große Tauchtiefen und eine lange Einsatzdauer. Bearbeitet wurde diese anspruchsvolle Aufgabe durch ein interdisziplinäres Konsortium, bestehend aus deutschen Firmen und Forschungseinrichtungen, mit Unterstützung des Bundesministeriums für Forschung und Entwicklung [55]. Dieses Unterwasserfahrzeug soll eine vorgegebene Mission durchführen und bei unvorhergesehenen Situationen (auftauchende Hindernisse, Ausfall von Baugruppen, starke Seeströmung) eigenständig handeln können. Die Auslegung der Leistungsparameter und der Systementwurf des Fahrzeuges resultieren aus den heutigen Anforderungen der maritimen Industrie und Wissenschaft. Eine Übersicht der wesentlichen Leistungsparameter des Fahrzeuges gibt Tabelle 1.1.

Tabelle 1.1: Leistungsparameter des AUVs DeepC

Parameter	Wert
Abmaße des Fahrzeuges	Länge 5800 mm Breite 2300 mm Höhe 1700 mm
Gewicht des Fahrzeuges	2,4 t
Missionstiefe	4000 m
Marschgeschwindigkeit	4 kn
Maximale Geschwindigkeit	6 kn
Einsatzdauer	40 h
Fahrstrecke	ca. 350 km
Nutzlast	250 kg
Energiegewinnung	Brennstoffzelle

Realisiert werden diese Eigenschaften u. a. durch die folgenden technischen Details (vergleiche die schematische Darstellung des Fahrzeuges mit seinen wichtigsten Antriebs- und Manövriereinrichtungen in Abbildung 1.1):

- Separate Nutzlast- und Fahrzeugkörper
- Redundanter Aufbau aller wichtigen Systeme und getrennte Unterbringung in den einzelnen Fahrzeugkörpern (1)
- Verwendung kohlefaserverstärkter Verbundstoffe (CFK) für die Druckkörper (2)
- Einsatz von Polymerelektrolytmembran - Brennstoffzellen (PEM-FC)
- Hochpräzises inertiales Navigationssystem (INS) [63]
- Austauschbarer Nutzlast-Container (5)

Seine guten Positionier- und Manövriereigenschaften im Stand und bei kleineren Geschwindigkeiten erhält das AUV durch die senkrecht und waagrecht eingesetzten Querstrahler (3). Bei höheren Geschwindigkeiten erfolgt die Steuerung des Fahrzeuges in einer Kombination von Querstrahler (3), Rudereinheiten (4) und Drehzahldifferenzen der Hauptantriebe (6).



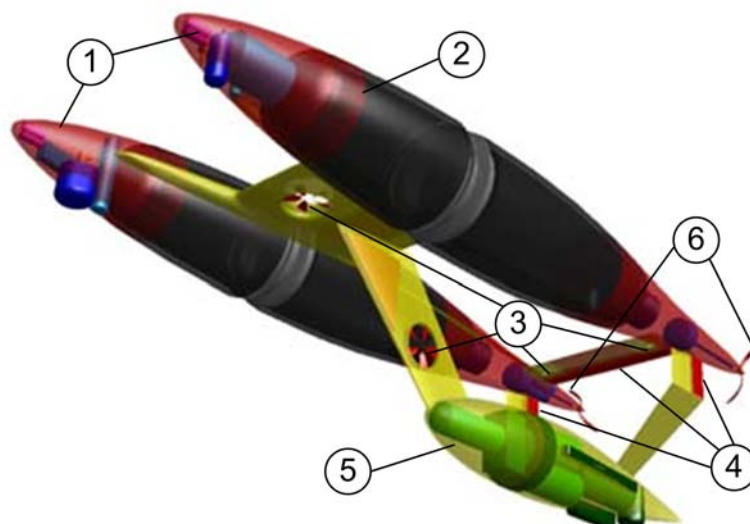


Abbildung 1.1: Strukturbild des AUVs DeepC [9]

## 1.1.2 Versuchsträger AUV M63

Für die praktische Überprüfung der im Forschungsprojekt „DeepC“ entwickelten Algorithmen zur aktiven Fahrzeugautonomie wurde neben dem AUV DeepC auch das AUV M63 aus der MARIDAN AUV 600 Serie der dänischen ATLAS Niederlassung ATLAS MARIDAN ApS [11] verwendet. Durch das eingesetzte modulare Führungskonzept (siehe Abschnitt 1.1.4) konnten die Algorithmen der einzelnen Module leicht an die fahrzeugspezifischen Schnittstellen und das Systemverhalten des M63 angepasst werden. Mit dem M63 stand ein praxiserprobtes AUV zur Verfügung, welches das Produkt einer jahrelangen kontinuierlichen Entwicklung der Firma MARIDAN ist. (Diese Firma wurde 1993 in Hoersholm/Dänemark gegründet und zählt zu den Pionieren auf dem Gebiet der AUV Entwicklung. Im Jahr 2004 wurde sie durch Akquisition von ATLAS übernommen.) Da die Seeversuche mit dem M63 parallel zur Entwicklung des Versuchsträgers DeepC durchgeführt wurden, konnten die entwickelten Algorithmen ohne Zeitverzug getestet und die dabei gewonnenen Erkenntnisse direkt in die DeepC Entwicklung einfließen. Tabelle 1.2 enthält die wichtigen Leistungsparameter des Fahrzeuges M63.

Tabelle 1.2: Leistungsparameter des AUVs M63

Parameter	Wert
Abmaße des Fahrzeuges	Länge 4500 mm Breite 2000 mm Höhe 710 mm
Gewicht des Fahrzeuges	1,7 t
Missionstiefe	600 m
Marschgeschwindigkeit	3 kn
Maximale Geschwindigkeit	5 kn
Einsatzdauer	7 h (Pb-Akku) 20 h (NiCd-Akku)
Fahrstrecke	ca. 40 km
Nutzlast	100 kg
Energiegewinnung	Pb-Akku NiCd-Akku

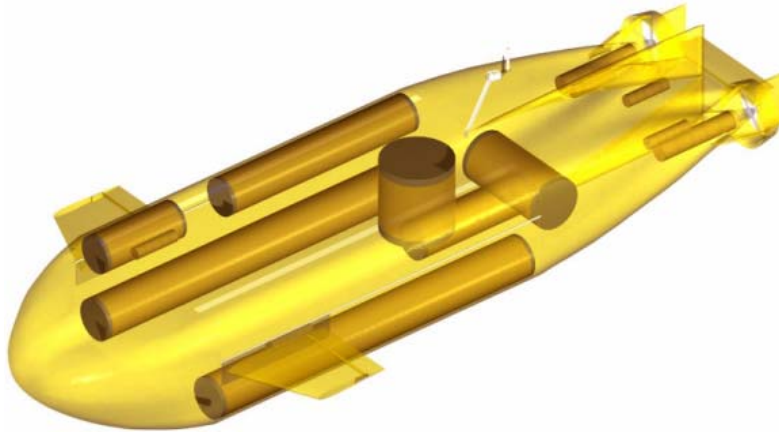


Abbildung 1.2: Strukturbild des AUVs M63 [9]

### 1.1.3 Einsatzmöglichkeiten für AUVs

Die Verlegung des ersten Unterwasserkabels zwischen Europa und Amerika um 1858 war ein finanzielles und technisches Wagnis. Es gingen zahlreiche Fehlversuche durch mangelnde Ausrüstung, falsche Technologie, schlechte Wetterbedingungen und fehlende Erfahrung voraus. In der heutigen Zeit sind diese Arbeiten dank der vorhandenen Technik und der ausgereiften Kabeltechnologie zur Routine geworden. Der Einsatz von Tiefseekabeln stellt eine echte Alternative zu Kommunikationssatelliten dar. So haben sie eine sehr geringe Ausfallwahrscheinlichkeit und können immense Datenmengen in einer geringeren Laufzeit transportieren [155]. Vor der Verlegung eines Unterwasserkabels sind genaue Informationen über die Trassenführung notwendig. Gebiete mit steilen Erhebungen, tiefgezogene Täler oder felsige Untergründe sind dabei zu vermeiden. In dieser Arbeitsphase kann ein AUV den Seeboden kartographieren oder eine optimale Route zur Trassenführung finden. Nach Fertigstellung der Kabelverlegungsarbeiten übernimmt das AUV Inspektionsaufgaben in vorgegebenen Serviceintervallen oder nach einem Seebeben. Bedenkt man, dass die mittlere Tiefe der Weltmeere 3790 m beträgt und die Kabel über lange Distanzen führen, ist in diesem Aufgabenbereich ein AUV eine finanzielle und technische Alternative zu einem ROV.

Heute gibt es teilweise detailliertere kartographische Informationen von der Oberfläche des Mondes und des Mars als vom Meeresboden. Solche kartographischen Informationen sind jedoch die Grundlage für die erfolgreiche Erforschung und Bewirtschaftung der Ozeane [107]. De Beers Marine setzt seit 1999 MARIDAN AUVs für die Seebodenvermessung zur Vorbereitung von Diamantenschürfungen ein [154]. Eine Kartographierung ist auch für die Festlegung der Souveränitätsansprüche von Küstenstaaten erforderlich. Laut des Seerechts-übereinkommens Artikel 77 übt ein Küstenstaat über seinen Festlandsockel souveräne Rechte zum Zweck seiner Erforschung und der Ausbeutung seiner natürlichen Ressourcen aus [18]. Die Definition des Festlandsockels ist durch eine Reihe von geographischen Eigenschaften (Festlandrand, Abhang, Anstieg, Gefällewechsel, Sedimentsgesteinsdicke) festgelegt, welche eine genaue Kartographierung verlangen. Für die Kartographierung der arktischen Gebiete bietet ein AUV die Möglichkeit die Seebodenvermessungen unter den meterhohen Eisschichten durchzuführen. Die kanadische Regierung sieht bei ihrem Arktischen Programm, bei dem bis 2012 alle Gebiete vermessen werden sollen, einen großen Handlungsbedarf für AUVs [148].

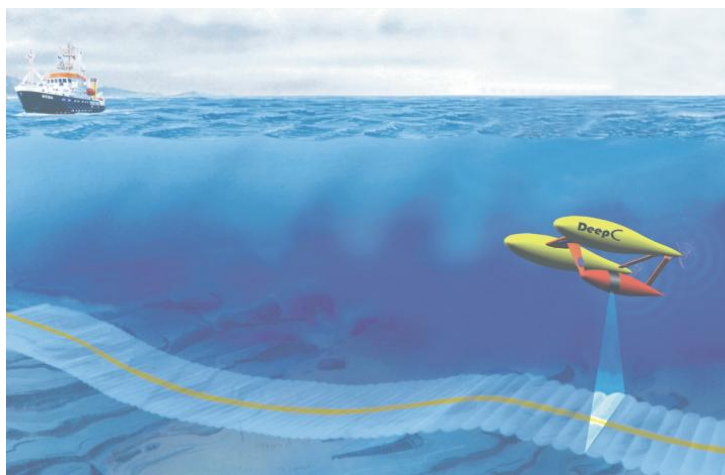


Abbildung 1.3: Vermessung des Seebodens [9]

Als Folge des enorm wachsenden Energiebedarfs hat die Erschließung von Öl- und Gasfeldern in Tiefseegebieten sowie das Interesse an der Gewinnung und dem Transport aus Seegebieten mit Eisbedeckung in den letzten Jahren stark zugenommen [64]. Der Transport des geförderten Erdöls und Erdgases erfolgt in den küstennahen Gebieten oder in Seegebieten mit Eisbedeckung durch Pipelines. Zur Bestimmung der Trassenführung und bei der Inspektion der Pipelines in Tiefseegebieten und unter Eis können AUVs effizient eingesetzt werden. In diesen Arbeitsumgebungen sind sie den ROVs im Aktionsradius und in der Mobilität durch ihre fehlende Kabelverbindung überlegen. Im Offshore-Bereich haben die HUGIN 3000 AUVs mit einer Tauchtiefe von 3000 m und einer Einsatzdauer von 60 h seit ihrer Einführung im Jahre 2000 mehr als 20000 Vermessungskilometer durchgeführt [96].

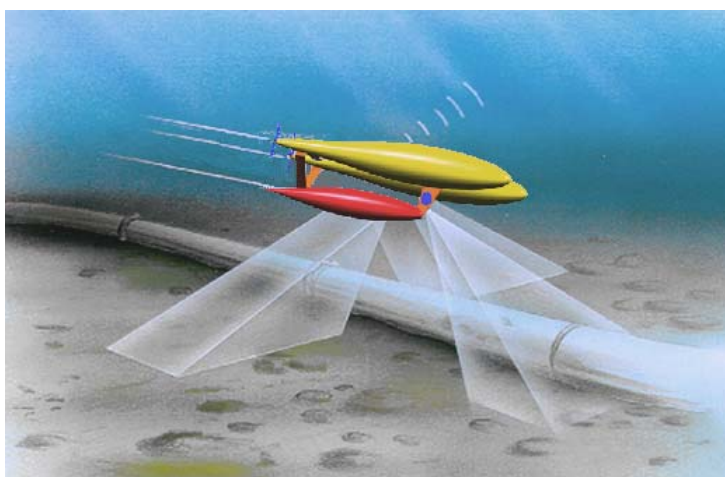


Abbildung 1.4: Inspektion einer Pipeline [9]

Bei der langfristigen Umstrukturierung der Energieversorgung und dem damit verbundenen Ausbau an erneuerbaren Energien, kommt der Errichtung von Offshore-Windparks eine zentrale Rolle zu. Solche Parks befinden sich in Küstennähe oder außerhalb der 12-Seemeilenzone (Ausschließlichen Wirtschaftszone (AWZ)) und können aus mehreren hundert Windkraftanlagen bestehen. Allein in Deutschland sind 29 Windparks (22 Nordsee / 7 Ostsee) in der AWZ beantragt (Stand Januar 2002) [32]. Ein mögliches Tätigkeitsfeld für ein AUV in einem Offshore-Windpark sind Inspektionsaufgaben im Bereich der Trägerkonstruktionen und der Energiekabel.



Abbildung 1.5: Inspektion eines Offshore-Parks [9]

Die Verschmutzung der Weltmeere steht in direktem Zusammenhang mit ihrer zunehmenden Nutzung. Solche Verschmutzungen entstehen durch gesunkene Schiffe mit gefährlicher Fracht, Tankerunfälle, Versenkungen von Abfallstoffen auf hoher See und Verunreinigungen bei der Erdölgewinnung [26]. Die AUVs können im Bereich des Umweltschutzes Such- und Überwachungsaufgaben übernehmen. So wurde im September 2004 das solarbetriebene AUV SAUV II zur Messung der gelösten Sauerstoffverteilung in der Greenwich Bay/USA eingesetzt [38].

### 1.1.4 Hierarchisches Führungskonzept des Fahrzeuges

Das in dieser Arbeit vorgestellte Modul ist Bestandteil eines Führungskonzeptes, welches in einer hierarchischen Dreiebenenarchitektur aufgebaut ist. Sie entspricht der von Brynes beschriebenen Rational Behavior Model (RBM) Architektur, die im AUV *NPS Phoenix* eingesetzt wird [29]. Bei diesem Konzept werden die zu lösenden Aufgaben innerhalb der *Strategischen Ebene*, der *Taktischen Ebene* und der *Ausführungsebene* aufgeteilt.

Vergleichbar ist dieses Hierarchiekonzept mit der Aufgabenorganisation auf einem U-Boot. Die *Ausführungsebene* entspricht dem Aufgabenbereich der Wachmannschaft. Zu ihren Aufgaben gehört die Steuerung des Schiffes sowie die Überwachung und Wartung der Maschinen und Sensoren. Geführt wird das Boot durch den kommandierenden Offizier. Sein Aufgabenbereich ist in der *Strategischen Ebene* angesiedelt und umfasst die Ausarbeitung einer Fahrtroute und das Festlegen von Missionsaufgaben. Überwacht und ausgeführt werden diese Aufgaben in der *Taktischen Ebene* durch den Wachoffizier. Er koordiniert ferner die Auswertung der Sonarbilder und der Navigationsdaten.

In Abbildung 1.6 sind die wichtigsten Softwaremodule innerhalb der Führungsarchitektur dargestellt. Im Normalfall wird ein erstellter *Missionsplan* abgearbeitet. Dieser Plan besteht aus einer Folge von Komplexmanövern, die spezielle Aufgaben (Abtauchen, Mäandern, GPS-Update) innerhalb der Mission beschreiben. Das Modul *Missionsplanabarbeitung* zerlegt die Komplexmanöver in einfache geometrische Gebilde, die Basismanöver. Aus dem aktuellen Basismanöver werden definierte Vorgaben zur Führung des Fahrzeuges in Form von Sollkurs, -tiefe, -bahn, -lage und -geschwindigkeit an den *Autopiloten* übergeben. Dieser Arbeitsmodus kann als vollautomatische Führung bezeichnet werden.





### 1.1.5 Aufbau und Aufgaben des Moduls FIS

Dieser Abschnitt stellt das Modul *Fahrzeugführung in Sondersituationen* im Überblick vor und beschreibt die Zusammenarbeit mit den anderen Modulen der Führungsarchitektur. Abbildung 1.7 zeigt die einzelnen Komponenten des Moduls *FIS*. Jede einzelne Komponente ist durch eine spezielle Aufgabe charakterisiert und wird in den nachfolgenden Ausführungen auch als Submodul bezeichnet.

Das Modul *FIS* wird durch den zyklischen Empfang der Navigationsdaten von der *Navigation* getriggert. Diese Navigationsdaten enthalten neben den Zustandsdaten des Fahrzeuges (Position, Geschwindigkeit, Lage) auch die Umweltdaten, wie Seeströmung und Abstand zum Seeboden. Nach dem Empfang der Navigationsdaten wird die Updateroutine des Moduls *FIS* durch eine sequenzielle Abarbeitung der einzelnen Submodule durchgeführt. Die von der *Objekterkennung* detektierten Objekte werden in einer Liste an das Softwaremodul *FIS* übergeben. Da alle ankommenden Datentelegramme in einer Warteschlange („Queue“) abgelegt werden, die nach dem FIFO Prinzip (engl.: first in, first out) arbeitet, werden erst nach der Durchführung der Updateroutine die empfangenen Objektdaten an die internen Datenstrukturen übergeben. Dieses Konzept ermöglicht den sicheren Zugriff auf gemeinsam genutzte Ressourcen und verhindert, dass während einer Updateroutine mit wechselnden Objektdaten gearbeitet wird. Das Submodul *Kollisionsüberwachung* überprüft während der Mission ständig eine Kollisionsmöglichkeit zwischen den detektierten Objekten und der Fahrtroute, welche aus dem aktuellen und den zukünftigen Basismanövern des Missionsplanes besteht (siehe Abschnitt 1.3.3). Dazu werden die Basismanöver der Fahrtroute gegen die gesamten detektierten Objekte auf Überschneidungen überprüft. Ist eine solche Möglichkeit gegeben, wird dies an das Modul *Missionsüberwachung* mit der Information der Kollisionsposition und des Kollisionszeitpunktes gemeldet.

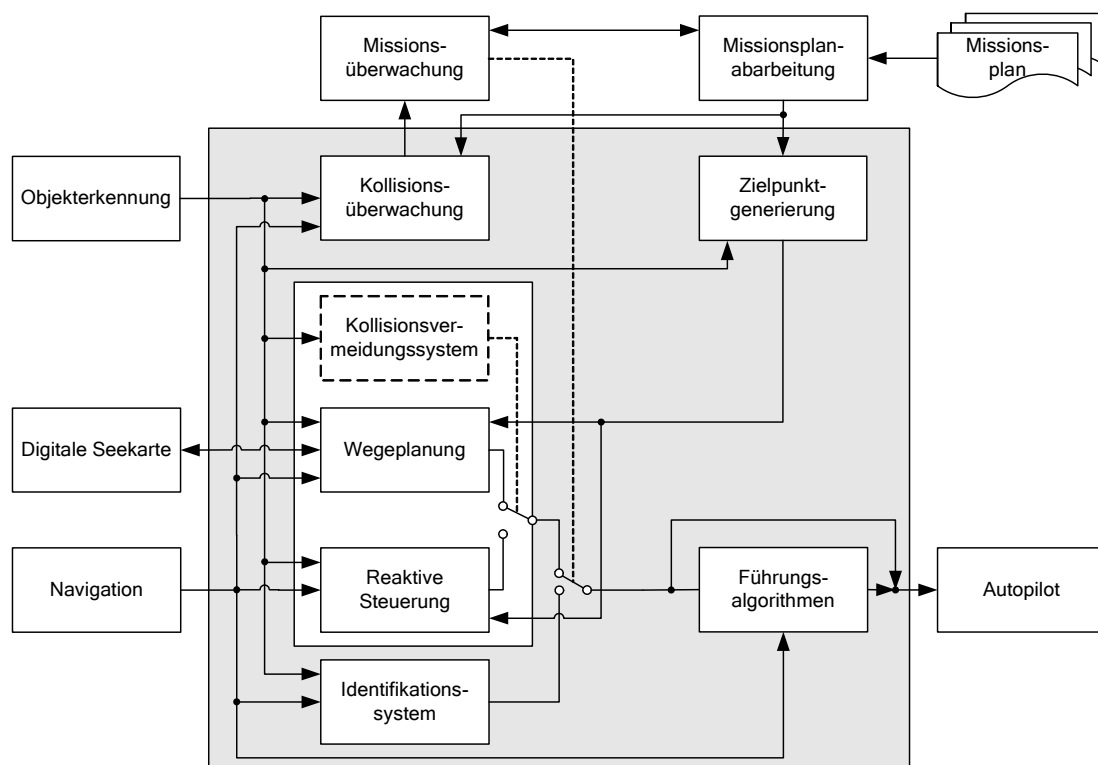


Abbildung 1.7: Struktur des Softwaremoduls *FIS*

Dieses Modul aktiviert in Abhängigkeit von der aktuellen Missionsaufgabe das *Kollisionsvermeidungssystem* oder das *Identifikationssystem* im Modul *FIS*. Das aktivierte System übernimmt dann die Führung des Fahrzeuges und übergibt die Steuerinformationen an das Submodul *Führungsalgorithmen* oder direkt an den Autopiloten. Die Steuerinformationen für die *Führungsalgorithmen* können Vorgaben für Solltrajektorien, anzufahrende Positionen oder der einzuhaltende Abstand zu einem Objekt sein.

Im Submodul *Führungsalgorithmen* werden diese Steuerinformationen durch die einzelnen Regelungen so aufbereitet, dass sie an den *Autopiloten* in Form von Sollwerten für Kurs, Tiefe und Geschwindigkeit übergeben werden. Diese universelle Schnittstelle zum Autopiloten erlaubt die Verwendung der Algorithmen des Moduls *FIS* auch für andere AUVs, unabhängig ihrer vorhandenen Aktuatorkonfiguration.

Die *Zielpunktgenerierung* berechnet während einer Sondersituation einen Rendezvouspunkt mit der abzufahrenden Route unter Verwendung des aktuellen Missionsplanes und der detektierten Objekte. Dieser Rendezvouspunkt ist der anzufahrende Zielpunkt des *Kollisionsvermeidungssystems*.

Für das *Kollisionsvermeidungssystem* des Fahrzeuges wurde eine Zweiebenen-Struktur favorisiert. Die obere Ebene verwendet zusätzlich zu den aktuellen Informationen des Sonars die gesammelten Hindernisdaten der bisherigen Mission sowie die Daten einer *Digitalen Seekarte*. Mit diesen Informationen findet eine *Wegeplanung* auf Basis geometrischer Graphen statt. Die Verfahren hierzu ermöglichen es, einen optimalen Weg nach definierten Vorgaben in einer kalkulierbaren Zeit zu ermitteln. Treten beim Abfahren des generierten Weges unvorhersehbare, plötzlich auftauchende Hindernisse auf, übergibt die Steuerung des *Kollisionsvermeidungssystems* die Führung des Fahrzeuges an die untere Ebene, die *Reaktive Steuerung*. Die *Reaktive Steuerung* reagiert auf die im Nahbereich des Sonars aufgefassten Hindernisse durch entsprechende reaktive Steuerkommandos, wodurch ein wegoptimales Umfahren der Objekte nicht immer möglich ist. Eine solche Forderung hat hier auch nicht die oberste Priorität. Aufgabe ist ein sicheres und schnelles Ausweichen zur Vermeidung einer Kollisionssituation. In der Aktivierungszeit der Ebene *Reaktive Steuerung* kann in der Ebene *Wegeplanung* der Routenplan unter Verwendung der neuen Objektinformationen modifiziert bzw. neu erstellt werden.

Bei der Durchführung einer Mission kann eine geforderte Aufgabe die Identifikation von Objekten sein. Diese Identifikation beinhaltet u. a. eine visuelle oder kartographische Erfassung des Objektes und/oder eine Bestimmung seiner physikalischen/chemischen Eigenschaften. Das Modul *Missionsüberwachung* übergibt dazu die Führung des Fahrzeuges an das Submodul *Identifikationssystem*. Hier werden, aus den Vorgaben der Objektgeometrie und den Anforderungen des Fahrzeuges während der Identifikation (Objektabstand, Ausrichtung zum Objekt, ggf. Fahrgeschwindigkeit), die Sollwerte für die *Führungsalgorithmen* und den unterlagerten Autopiloten gebildet. In Abhängigkeit der Objektgröße werden die zwei Arbeitsmodi *Ausrichten auf ein Objekt* und *Umfahren eines Objektes* unterschieden. Nach Abschluss einer Identifikation wird ein anzufahrender Zielpunkt auf der aktuellen Missionsroute durch das Submodul *Zielpunktgenerierung* ermittelt. Dieser Zielpunkt wird dann durch das intern aktivierte *Kollisionsvermeidungssystem* angefahren. Das selbstständige Umschalten in den Ausweichmodus erfolgt auch, wenn ein Objekt schon als identifiziert gekennzeichnet wurde.

## 1.2 Anforderungen, Vorgaben und Bedingungen

Durch die Verwendung des Softwaremoduls *Fahrzeugführung in Sondersituationen* in einem AUV ergeben sich eine Reihe von praxisrelevanten Anforderungen an die eingesetzten Verfahren und Algorithmen, welche unter den hard- und softwaretechnischen Projektvorgaben und Arbeitsbedingungen einzuhalten sind (siehe Abbildung 1.8). Die nachfolgenden Abschnitte beschreiben die wichtigsten Anforderungen, Vorgaben und Bedingungen im Überblick. Auf diese Ausführungen wird bei der Behandlung der einzelnen Submodule in dieser Arbeit Bezug genommen. So wurden bei den durchgeführten Literaturrecherchen besonders Ansätze im Unterwasserbereich bzw. mit einer praktischen Realisierung untersucht und nach den vorgegebenen Anforderungen bewertet. Die Entwicklung der Algorithmen erfolgte unter Einbeziehung der in der Praxis vorzufindenden Arbeitsbedingungen. Die einzuhaltenden Parameter bei den Tests (Updatezeiten, Rechenleistung, Objektanzahl) wurden aus den hard- und softwaretechnischen Vorgaben abgeleitet.

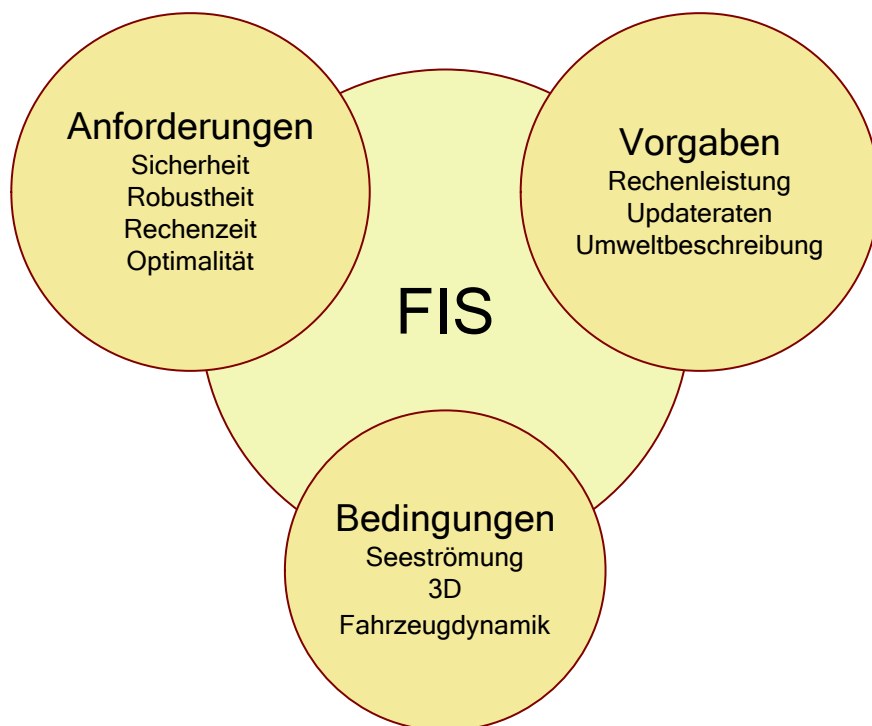


Abbildung 1.8: Diagramm des Moduls *FIS*

### 1.2.1 Anforderungen

Die Anforderungen an das Modul *FIS* lassen sich durch vier Kernmerkmale klassifizieren, die eine vorgegebene Wertigkeit innerhalb einer Prioritätsliste besitzen (siehe Abbildung 1.9). In dieser Liste hat die *Sicherheit* des Fahrzeuges und seine umgebende Umwelt höchste Priorität. Eine Kollision mit Objekten ist auf jeden Fall zu vermeiden. Übernimmt das Modul *FIS* die Fahrzeugführung, dürfen die generierten Steuerkommandos nicht zu einem unkontrollierten bzw. instabilen Fahrzeugverhalten führen. Des Weiteren muss das auf der Rechereinheit laufende Programm echtzeitfähig sein und darf keine schwerwiegenden Softwarefehler besitzen sowie nicht zur Blockierung oder gar zum Absturz des Gesamtsystems führen.



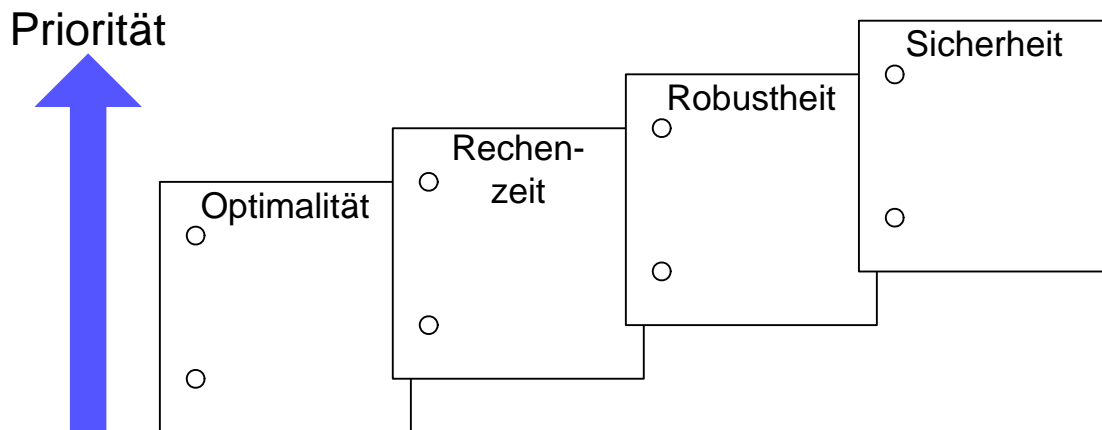


Abbildung 1.9: Prioritätsliste

Die entworfenen Algorithmen müssen eine notwendige *Robustheit* gegenüber ungenauen Navigations- und Objektdaten besitzen. So sind Ausweichverfahren gefordert, welche auch bei stochastischen Variationen der Objektdaten infolge der entfernungsabhängigen Detektionsgüte des Sonars zuverlässig arbeiten. Die Regelungsalgorithmen müssen für Führung und Störung ausgelegt sein und gegenüber dem Variieren des Fahrzeugverhaltens in bestimmten Bereichen unempfindlich bleiben.

Alle eingesetzten Verfahren sollten neben der Grundforderung nach Echtzeitfähigkeit auch eine geringe *Rechenzeit* besitzen. Diese Forderung resultiert aus der Tatsache, dass auf den einzelnen Rechereinheiten mehrere Softwaremodule in eigenständigen Prozessen arbeiten. Einige Module können sehr rechenintensiv sein, was unter bestimmten Bedingungen zu einer Vollausslastung des Prozessors und damit zur Verletzung der Echtzeitanforderung führt. In Liebezeit [103] wurde deshalb schon in der Spezifikationsphase mit dem Entwurfsprogramm MLDesigner eine erste Performanceanalyse zur Festlegung der notwendigen Rechenleistung, der Anzahl der Rechereinheiten und der darauf laufenden Module durchgeführt.

Eine primäre Forderung bei der Wegeplanung ist die *Optimalität* der ermittelten Routen. So soll die Route unter Berücksichtigung einer vorhandenen Seeströmung in einer kurzen Zeit abgefahren werden. Die *Optimalität* hat die niedrigste Wertigkeit in der definierten Prioritätsliste. Dies erklärt sich daraus, dass der gefundene Weg in einem vorgegebenen Zeitrahmen ermittelt werden muss (*Rechenzeit*) und unempfindlich gegenüber Variationen der Objektgeometrie sein soll (*Robustheit*). Die Einhaltung der Zeitforderung kann durch eine gröbere Nachbildung der Umwelt in den geometrischen Graphen erreicht werden. Einer Variation der Objektdaten wird durch die Vorgabe eines Sicherheitsbereiches um die einzelnen Objekte Rechnung getragen. Beide Maßnahmen senken dabei die geforderte *Optimalität*. An diesem Beispiel ist zu sehen, dass immer ein Kompromiss zwischen den einzelnen Anforderungen unter Beachtung ihrer Priorität bei der Entwicklung der Verfahren gefunden werden muss.

## 1.2.2 Hardwaretechnische Vorgaben

### Computer

Das Softwaremodul *FIS* arbeitet als eigenständiger Prozess auf einem Single Board Computer des Typs VP7 mit einem Pentium III 500 MHz Prozessor der Firma SBS Technologies [131]. Sein Prozessor- und Speicherbedarf beträgt 25 % der Rechereinheit.

## Fahrzeug

In den Abschnitten 1.1.1 und 1.1.2 wurden die AUVs, auf denen das Modul *FIS* arbeitet, kurz vorgestellt. Die beiden Fahrzeuge unterscheiden sich in ihren Aktuatorkonfigurationen, Massen, Trägheitsmomenten und bedingt durch ihre Geometrie in ihren hydrodynamischen Widerständen und Momenten. Daraus ergibt sich ein fahrzeugspezifisches Systemverhalten, welches auf der Basis eines physikalischen Modells beschreibbar ist [48, 58] und beim Regelungsentwurf berücksichtigt werden muss. Die Fahrzeuge besitzen ein nichtholonomes Bewegungsverhalten. Dieses Verhalten resultiert in der vertikalen Ebene aus der Verwendung der Fahrzeugnicklage zur Erreichung einer vorgegebenen Solltiefe. So kann eine Tiefenänderung nur mit einer gleichzeitigen horizontalen Positionsveränderung einhergehen. Die Fahrzeugnicklage ergibt sich aus dem eingestellten Ruderausschlag, der Fahrtgeschwindigkeit, der Ruder- und Fahrzeuggeometrie und der Lage zwischen Masse- und Auftriebschwerpunkt. In der horizontalen Ebene erfolgt die Einnahme einer Position durch Drehung des Fahrzeuges, wobei sich die Drehrate aus dem Ruderausschlag und/oder der Drehzahldifferenz der Hauptantriebe, der Fahrtgeschwindigkeit und der Ruder- und Fahrzeuggeometrie bestimmt. So entsteht ein Drehkreis, wenn ein Fahrzeug eine Richtungsänderung durchführt.

## Sonar

Als Sensor zur Umwelterfassung wird das Sonarsystem Super SeaKing DFS der Firma Tritech [143] eingesetzt. Dieses Sonar arbeitet als vorausschauendes Sektor Scanning System. Dabei wird die Umwelt durch eine sequentielle Folge von Schallimpulsen (Pings) abgerastert. Nach Ablauf der voreingestellten Laufzeit  $\Delta t_{\max}$  zwischen dem ausgesendeten Ping und einem empfangenen Echosignal wird das Sonar mechanisch um die Winkelschrittweite  $\Delta\alpha$  weitergedreht. So findet ein kontinuierliches Hin- und Herschwenken des Sonars zwischen seinen Sektorgrenzen  $\pm\alpha_{\max}$  statt, wodurch nach jedem Schwenk ein Sonarbild des gesamten Sektors entsteht. Die Zeit  $t_B$  zur Generierung eines solchen Bildes ist vom voreingestellten Auffassungsbereich  $r_{\max}$ , der Sektorgrenze  $\alpha_{\max}$  und der Winkelschrittweite  $\Delta\alpha$  abhängig und kann durch die folgende Gleichung bestimmt werden:

$$\begin{aligned} c &= 1500 \frac{\text{m}}{\text{s}} \quad (\text{Schallgeschwindigkeit im Wasser}) \\ t_{\max} &= \frac{2 \cdot r_{\max}}{c} \\ t_B &= \left\lceil \frac{2 \cdot \alpha_{\max}}{\Delta\alpha} \right\rceil \cdot t_{\max} = \left\lceil \frac{2 \cdot \alpha_{\max}}{\Delta\alpha} \right\rceil \cdot \left( \frac{2 \cdot r_{\max}}{c} \right) \end{aligned} \quad (1.1)$$

Abbildung 1.10 stellt die Zeit zur Generierung eines Sonarbildes in Abhängigkeit des Auffassungsbereiches  $r_{\max}$  und der Sektorgrenze  $\alpha_{\max}$  bei einer Winkelschrittweite von  $\Delta\alpha = 0.9^\circ$  dar. Für den realen Betrieb wurde ein Sektor von  $\pm 30^\circ$  und ein maximaler Auffassungsbereich von 50 m definiert, wodurch die Zeit für die Erzeugung eines Sonarbildes nach Gleichung (1.1) 4.47 s beträgt. Aus diesem Sonarbild werden im Modul *Objekterkennung* die einzelnen Objekte detektiert. So ergibt sich die Updatezeit zur Bereitstellung einer neuen Objektliste  $\Delta t_{\text{Objekt}}$  aus der Generierungszeit eines Sonarbildes  $t_B$  und der anschließenden Bearbeitungszeit des Sonarbildes. Da zur Verifikation der im Sonarbild detektierten Objekte mindestens ein zweites Bild ausgewertet werden muss [79], ergibt sich noch eine zusätzliche Verzögerung in der Größe der Updatezeit  $\Delta t_{\text{Objekt}}$  für die Bereitstellung neu detektierter Objekte.

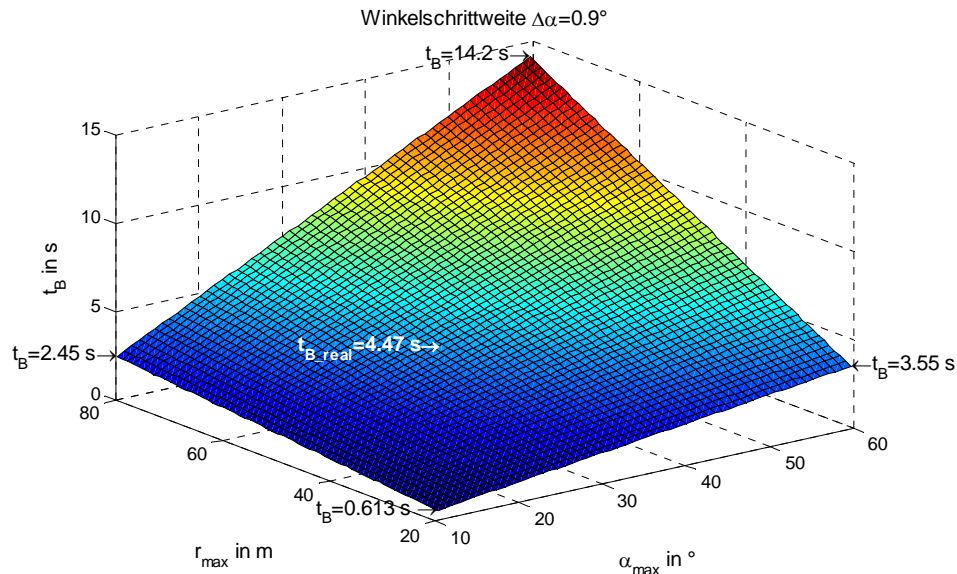


Abbildung 1.10: Verlauf der Zeit für die Erzeugung eines Sonarbildes  $t_B = f(r_{\max}, \alpha_{\max})$  bei  $\Delta\alpha = 0.9^\circ$

### 1.2.3 Softwaretechnische Vorgaben

Dieser Abschnitt stellt die wichtigsten softwaretechnischen Vorgaben in einem kurzen Überblick vor. Bei der Definition dieser Vorgaben wurden auch die im Abschnitt 1.2.1 beschriebenen Anforderungen einbezogen. So findet sich der Sicherheitsaspekt in den nachfolgenden Vorgaben wieder.

#### **Softwareentwicklung**

Für die Softwareentwicklung werden Methoden der objektorientierten Analyse (OOA, *Object Oriented Analysis*) und des objektorientierten Entwurfes (OOD, *Object Oriented Design*) eingesetzt. Eine gut durchgängige Strukturierbarkeit zwischen den Phasen Analyse und Entwurf wird durch die Verwendung der selben Notation, der UML (*Unified Modeling Language*), erreicht [12]. Sie ist die Modellierungssprache von Rational Rose, welches als CASE-Tool für die Analyse, den Entwurf, die Implementierung und die Dokumentation eingesetzt wird [24], [82]. Die Vorteile der objektorientierten Softwareentwicklung liegen im modularen Aufbau des Codes durch die Nutzung des Klassenkonzeptes und die damit verbundene einfache Wartbarkeit und Wiederverwendbarkeit sowie eine leichte Erweiterbarkeit der Softwaremodule.

#### **Programmierung**

Für die Programmierung der Algorithmen wird C++ verwendet. Die eingesetzte Entwicklungsumgebung ist der Borland C++ Builder 6.0 [23]. Die Codierung erfolgt unter Berücksichtigung der vorgegebenen Codierrichtlinien [10] und -regeln [127].

#### **Kommunikation**

Die Kommunikation mit den anderen Modulen erfolgt über CORBA (Common Object Request Broker Architecture). CORBA stellt einen objektorientierten Ansatz für die Entwicklung verteilter Anwendungen dar. Die CORBA-Anwendungen können mit unterschiedlichen Programmiersprachen erstellt und auf verschiedenen Zielplattformen ausgeführt werden [22]. CORBA bietet eine Reihe von Standarddiensten an, die zu einer hohen Servicequalität und Kommunikationssicherheit führen. Durch den zyklischen Empfang der Navigationsdaten wird das Modul FIS alle 100 ms getriggert.

## ***Bibliotheken***

Bei der programmtechnischen Umsetzung werden vorhandene bzw. standardisierte Bibliotheken genutzt (BOOST [20], STL [99], XERCES [156]). Standardisierte Bibliotheken sind umfassend geprüft und werden ständig weiterentwickelt. Die Wahrscheinlichkeit von Softwarefehlern ist deshalb sehr gering. Des Weiteren ist bei der Verwendung von Standardbibliotheken der Einarbeitungsaufwand für andere Entwickler geringer, eine Dokumentation ist vorhanden und es existiert eine Vielzahl von Applikationsbeispielen und Nutzererfahrungen. Da die Häufigkeit von Softwarefehlern direkt mit der Programmgröße korreliert [53] und die Funktionalitäten der Bibliotheken nicht selbst programmiert werden brauchen, führt ihre Verwendung auch zu einer Erhöhung der Softwarezuverlässigkeit.

## **1.2.4 Arbeitsbedingungen**

### ***Umweltbedingungen***

Im Vergleich zu landgeführten mobilen Systemen kann im Arbeitsraum eines Autonomen Unterwasserfahrzeuges eine Vielzahl von widrigen Umweltbedingungen existieren. Diese sind absolute Dunkelheit, hoher Wasserdruck, hydrothermische Quellen am Grund der Tiefsee (Smoker), aufsteigende Gasblasen, Wirbel und Strömungen.

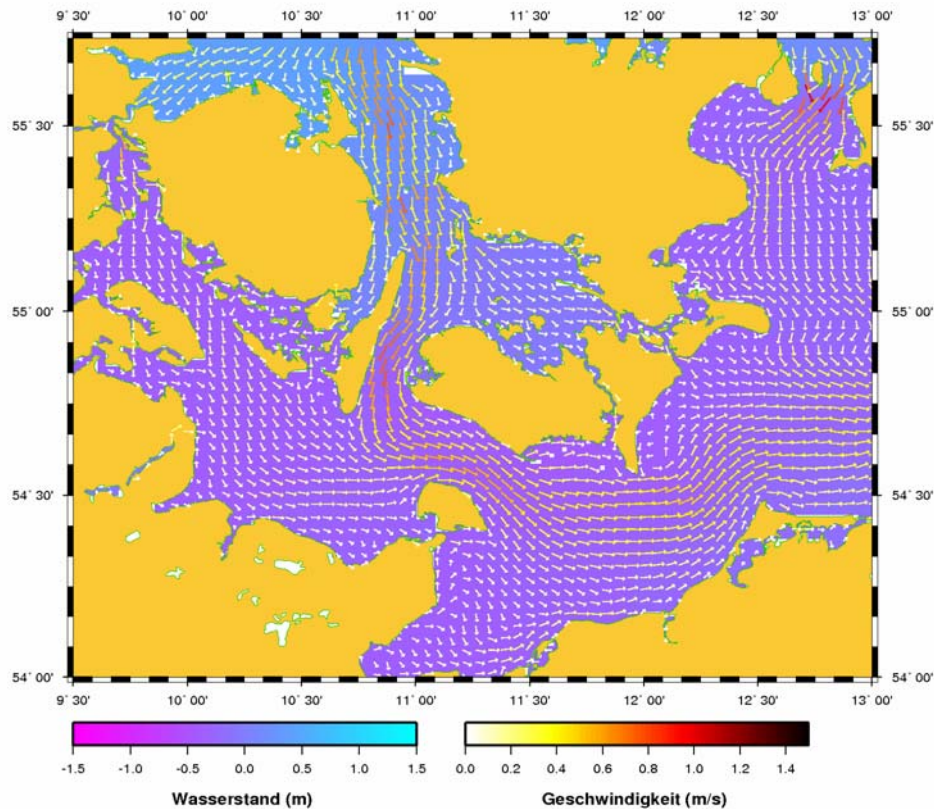
Strömungen können ihre Ursachen in den Gezeiten, dem Wind, der Gravitation oder der Wassertemperaturdifferenz haben. Die Berücksichtigung der vorhandenen Strömung ist für ein sicheres und optimales Manövrieren eine wichtige Voraussetzung. So sollten Gebiete mit einer großen Gegenströmung bei der Wegeplanung gemieden werden. Die örtliche Strömung am Fahrzeug kann durch ein Doppler Velocity Log (DVL) [140] oder durch eine Schätzung mittels eines Beobachtermodells [48] ermittelt werden. Für die Bestimmung bzw. Vorhersage eines dreidimensionalen Strömungsfeldes werden operationelle numerische Simulationsmodelle [40] eingesetzt. Diese Modelle bestimmen den Strömungsverlauf in einem hydrodynamischen Modell unter Einbeziehung der Wetter-, Wellen- und Gezeitenvorhersage. Solche berechneten Strömungsverläufe des gesamten Operationsgebietes können bei der Missionsvorbereitung, der Missionsumplanung oder der *Wegeplanung im Kollisionsvermeidungssystem* verwendet werden. Diese Strömungsverläufe könnten im Vorfeld einer Mission oder während einer Mission unter zusätzlicher Einbeziehung der aktuellen Strömung berechnet werden. Abbildung 1.11 zeigt das mit dem Simulationsprogramm BSHcmod [40] erstellte Strömungsfeld der westlichen Ostsee.

### ***Objekte***

Als Objekte werden alle Körper angesehen, die dem AUV bei einer Kollision gefährlich werden können. Dazu zählen geographische Hindernisse (Smoker, Wände), technische Bauten (Träger, Kaimauern) und technische Altlasten (Container, Minen, Schiffwracks). Kleine im Wasser treibende Objekte mit einer Ausdehnung kleiner 0.5 m stellen keine Gefahr für das AUV dar. Bewegliche Objekte sind Lebewesen, die dem AUV selbständig ausweichen [79].

### ***Arbeitsraum***

Die Fahrzeuge können sich in einem dreidimensionalen Raum bewegen. Einschränkungen sind durch die o.g. Objekte und einen einzuhaltenden Sicherheitsabstand zum Meeresboden gegeben.



**BSH-Zirkulationsmodell      Modellschicht 1**

**Stroemungen am 07.02.2006 06:00 MEZ**

BSH/S.Dick

Abbildung 1.11: Strömungsmodell der westlichen Ostsee [40]

## 1.3 Beschreibung der Datenstrukturen

Die in dieser Arbeit verwendeten Datenstrukturen sollen in den nachfolgenden Abschnitten kurz vorgestellt werden.

### 1.3.1 Objektmodellierung

Nach jedem Sonar-Update findet im Modul *Objekterkennung* eine Auswertung des Sonarbildes zur Detektion der im Operationsgebiet befindlichen Objekte statt. Sie werden anschließend an das Modul *FIS* in einer Objektliste übergeben. Das Modul *Objekterkennung* bildet die detektierten Objekte als elliptische Zylinder nach. Die Symmetrieachse des Zylinders verläuft dabei parallel zur z-Achse. Diese geometrische Form lässt sich leicht aus dem Sonarbild generieren und charakterisiert durch wenige geometrische Parameter (Hauptachse  $a_{\text{Ellipse}}$ , Nebenachse  $b_{\text{Ellipse}}$  und Rotationswinkel  $\theta_{\text{Ellipse}}$ , Höhe des Zylinders  $h$  und Zentrumsposition  $x_{\text{Ellipse}}$ ) eine Vielzahl von praktischen Hindernissen. Abbildung 1.12 beschreibt die Parameter des elliptischen Zylinders. Die elliptische Beschreibungsform des Objektes wird auch in einer Vielzahl von anderen Arbeiten verwendet [71, 81, 122]. In [56] werden elliptische Kreisbögen zur Nachbildung von Pipelines bei der Auswertung von Unterwasserbildern eingesetzt.



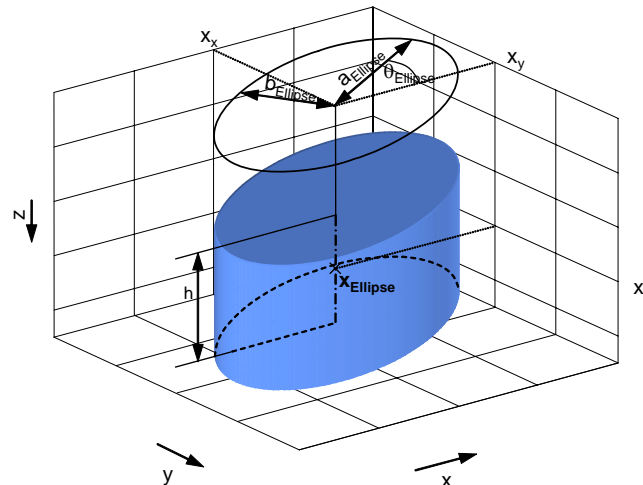


Abbildung 1.12: Parameter eines elliptischen Zylinders

Die elliptische Form gestattet auch eine Beschreibung von komplexen Objektstrukturen. Abbildung 1.13 zeigt die Verwendung der elliptischen Zylinder für die Nachbildung des Küstenverlaufes bei Camaret sur Mer in der Bretagne [33]. Für den gesamten Küstenverlauf und zur Beschreibung der Hafenanlagen wurden lediglich 12 Objekte definiert. Geradlinige Küstenverläufe oder Kaimauern lassen sich einfach durch das Ziehen der Hauptachse  $a_{\text{Ellipse}}$  auf die Länge des Objektes und das Stauchen der Nebenachse  $b_{\text{Ellipse}}$  beschreiben. Bei Sperrgebieten, welche die Form eines n-Eckes besitzen, werden n Ellipsen zur Abdeckung der einzelnen Kanten verwendet. So ist es möglich, aus den Daten einer digitalen Seekarte die Küstenverläufe oder auch gesperrte Gebiete durch elliptische Zylinder nachzubilden, um diese während der Mission für das *Kollisionsvermeidungssystem* zu verwenden.

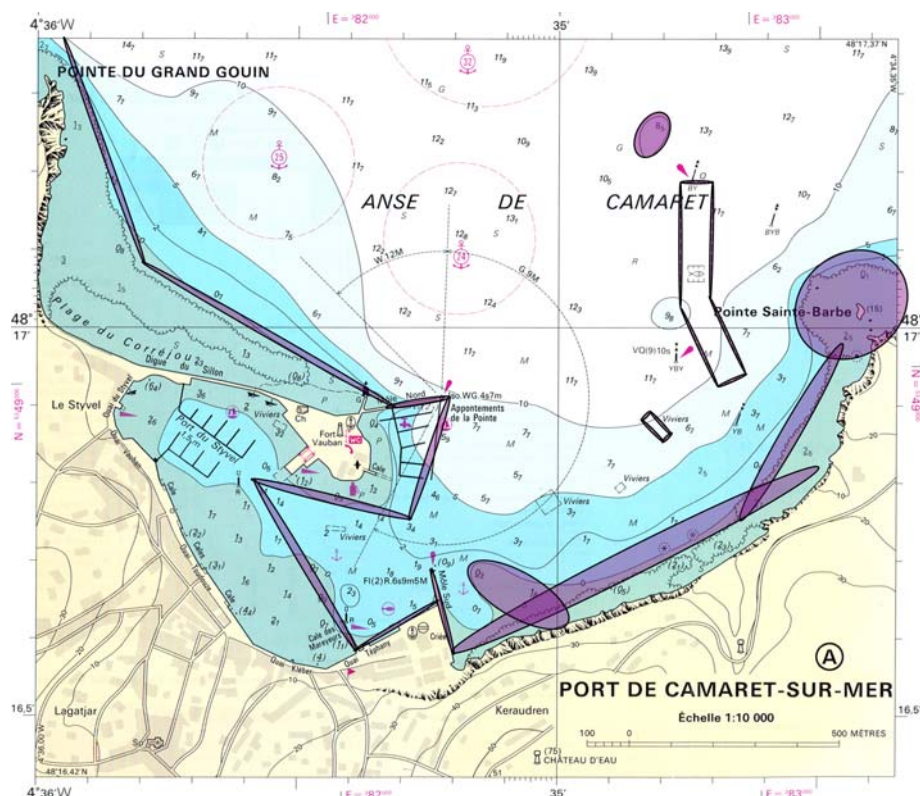


Abbildung 1.13: Beschreibung der Küstengebiete durch elliptische Formen

### 1.3.2 Bereichsdefinitionen

In Abhängigkeit von der Verwendung der Objekte in den einzelnen Submodulen des Moduls *FIS* werden zwei Bereiche um die Objekte definiert (siehe Abbildung 1.14). Durch die Abmaße des Fahrzeuges, seine mögliche Ablage zur Sollbahn, das Tangieren der berechneten optimalen Ausweichbahn mit den Objekten sowie auf Grund einer Varianz der Objektpositionen und -geometrien müssen die detektierten Objekte für ihre Verwendung im Submodul *Kollisionsvermeidungssystem* um einen *Sicherheitsbereich* erweitert werden. Dies erfolgt durch die Vergrößerung der Haupt- und Nebenachse der elliptischen Grundfläche sowie die Erhöhung der zylindrischen Objektform um einen vorgegebenen *Sicherheitsabstand*. So wird sichergestellt, dass das Fahrzeug ein Objekt in einem vorgegebenen Sicherheitsabstand umfährt und nicht mit ihm kollidiert. Ein zweiter Bereich, welcher um die Objekte gelegt wird, kommt bei den Submodulen *Kollisionsüberwachung* und *Zielpunktgenerierung* zum Einsatz. Dieser Bereich hat zwei Funktionen. Da Objekte nur identifiziert werden, wenn sie bei einer Kollisionsüberprüfung als mögliche Kollisionskandidaten erkannt wurden, kann durch eine Vergrößerung der Objektgeometrie um einen *Beobachtungsbereich* Einfluss auf ihre Identifikation genommen werden. Das Submodul *Zielpunktgenerierung* ermittelt anhand einer Kollisionserkennung mit den Objekten einen Zielpunkt (siehe Kapitel 7), wobei die verwendeten Objektmaße nicht kleiner als bei der *Kollisionsüberwachung* sein dürfen. Sonst kann es passieren, dass ein Ausweichmanöver durch eine mögliche Kollision aktiviert wurde und die *Zielpunktgenerierung* durch die kleineren Objektmaße keine Kollision ermitteln konnte, wodurch der anzufahrende Zielpunkt vor dem Objekt liegt. Eine weitere Funktion des *Beobachtungsbereiches* liegt in der Positionierung des Zielpunktes begründet. Der anzufahrende Zielpunkt sollte weit genug vom Objekt entfernt liegen, um ein sicheres Einfahren auf den Missionsplan zu gewährleisten. So richtet sich der definierte *Beobachtungsbereich* auch nach der Manövrierfähigkeit des Fahrzeuges und schließt immer den definierten *Sicherheitsbereich* mit ein. Bei den Darlegungen in den einzelnen Kapiteln wird davon ausgegangen, dass die Objektgeometrie um den jeweiligen Bereich vergrößert wurde.

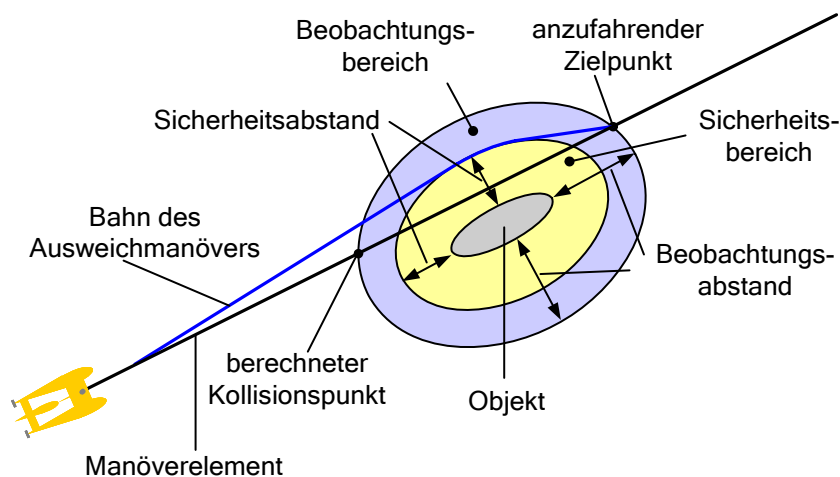


Abbildung 1.14: Definition der Bereiche um ein Objekt

### 1.3.3 Missionsplan

Ein Missionsplan setzt sich aus einer Folge von Manövern zusammen. Diese Manöver sind in *Komplexmanöver* und *Übergangsmanöver* unterteilt. Komplexmanöver erfüllen eine spezielle Aufgabe (Ab- und Auftauchen, Mäander, GPS-Update) innerhalb des Missionsplanes.

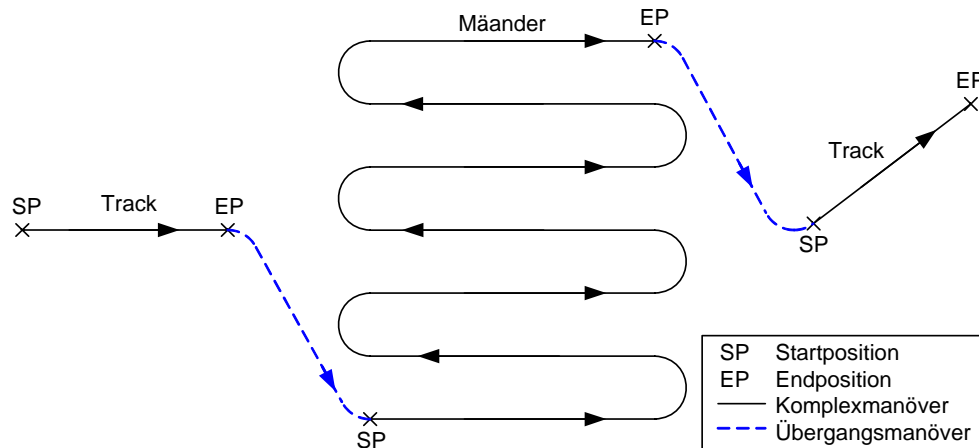


Abbildung 1.15: Missionsplan

Die *Übergangsmanöver* dienen als Verbindung zwischen End- und Startposition zweier *Komplexmanöver*. Abbildung 1.15 zeigt einen Missionsplan mit diesen beiden Manövertypen. Die *Komplex-* und *Übergangsmanöver* setzen sich aus Basismanövern zusammen. Diese Basismanöver stellen einfache geometrische Linienverläufe dar, welche durch eine Start- und Endposition, geometrische Daten zur Beschreibung ihrer Form und Sollgeschwindigkeitsvorgaben charakterisiert sind. Nach Fossen [58] umfasst die Lösung einer Manöveraufgabe zwei Teile:

*Geometrische Aufgabe:* Führung des Fahrzeuges auf der vorgegebenen Solltrajektorie.

*Dynamische Aufgabe:* Steuerung des Fahrzeuges nach den Sollgeschwindigkeitsvorgaben.

Abbildung 1.16 zeigt die vier verwendeten Basismanöver *Strecke*, *Kreisbogen*, *Kreis* und für das Auf- und Abtauchen die *achtförmige Spirale* (*Kreuzspirale*). Die Basismanöver bilden somit die Grundbausteine eines Missionsplanes. Aus den Komplex- bzw. Übergangsmanövern wird vom Modul *Missionsplanabarbeitung* das aktuelle Basismanöver herausgelöst und an den *Autopiloten* übergeben (siehe Abbildung 1.6). Dort werden die beiden oben genannten Manöveraufgaben unter Vorgabe der geometrischen und dynamischen Parameter durch entsprechende Regelungen gelöst. Abbildung 1.17 zeigt den Aufbau eines Missionsplanes mit Basismanövern aus dem in Abbildung 1.15 dargestellten Missionsplan.

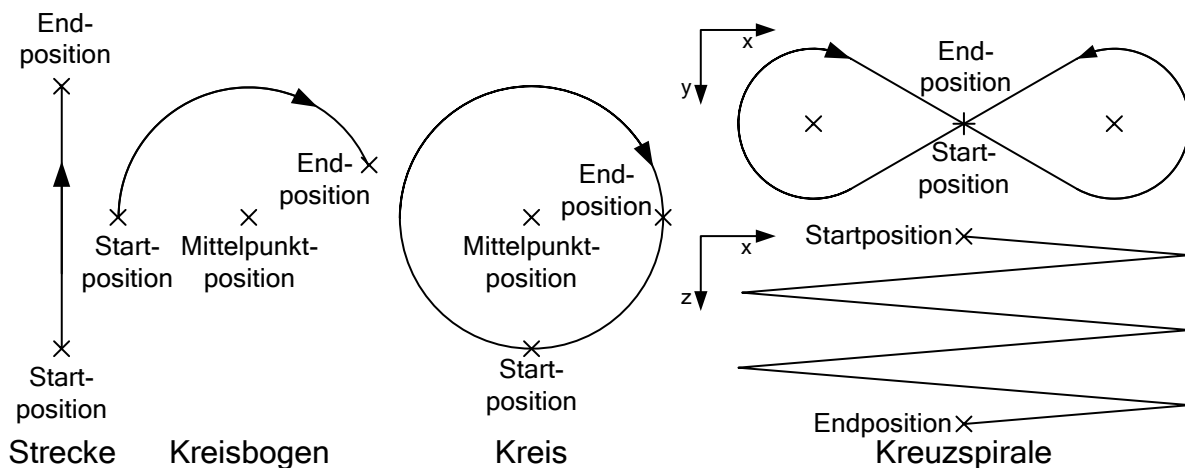


Abbildung 1.16: Basismanöver



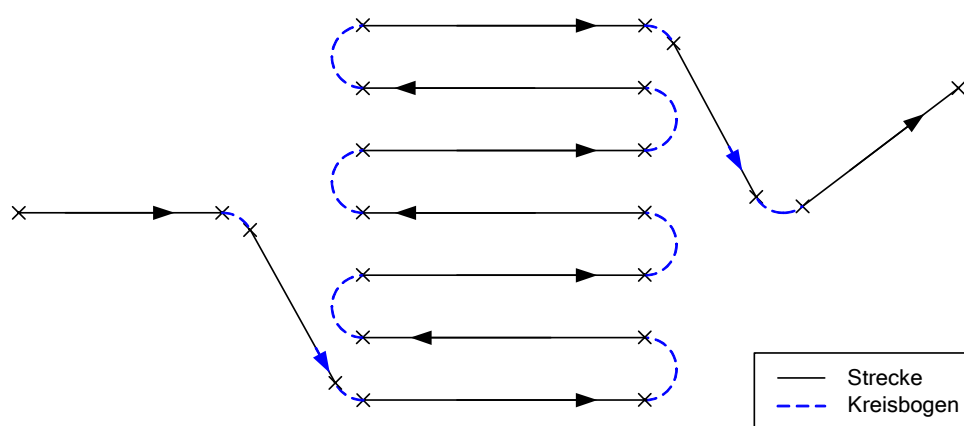


Abbildung 1.17: Basismanöverplan

## 1.4 Zielsetzung, Aufbau und Beitrag der Arbeit

Die in den einzelnen Kapiteln dieser Arbeit vorgestellten Verfahren wurden auf der Basis der im Abschnitt 1.2 dargestellten Anforderungen, Vorgaben und Bedingungen innerhalb einer dreijährigen Projektlaufzeit, vom Entwurf bis zu den abgeschlossenen Tests der Algorithmen bei Seeversuchen, entwickelt.

Die einzelnen Submodule werden separat in den Kapiteln 2 bis 8 behandelt. Diese Kapitel sind weitestgehend unabhängig voneinander aufgebaut. Sie umfassen neben der Beschreibung des Verfahrens eine einführende Darstellung der jeweiligen Teilaufgabe, eine Vorstellung von bekannten Ansätzen sowie eine kurze Zusammenfassung am Ende jedes Kapitels. Dieser Aufbau gestattet dem interessierten Leser einen einfachen Zugang und eine leichtere Verwendung von Teilresultaten aus dieser Arbeit. Abbildung 1.18 zeigt den Aufbau der Arbeit.

Kapitel 2 stellt das Verfahren zur *Kollisionsüberwachung* vor. Der Schwerpunkt ist hierbei die Vorstellung eines effizienten Verfahrens mit der Untersuchung seiner Leistungsfähigkeit in Bezug auf die Objekt- und Basismanöveranzahl.

Ausgehend von den Anforderungen an ein *Kollisionsvermeidungssystem* werden in Kapitel 3 mögliche Architekturen vorgestellt und diskutiert.

Kapitel 4 beschreibt das Konzept zur *Wegeplanung* auf Basis eines geometrischen Graphen. Dabei werden mehrere Ansätze zur Erstellung eines solchen Graphen vorgestellt und auf ihre Leistungsfähigkeit untersucht.

In Kapitel 5 wird, ausgehend von den Verfahren der künstlichen harmonischen Dipol-Potentiale [71], ein neues Verfahren zur Erzeugung von Gradientenlinien durch geometrische Konstruktion für eine *Reaktive Steuerung* zur Kollisionsvermeidung vorgestellt.

Die Algorithmen zur Führung des Fahrzeuges bei Identifikationsaufgaben werden im Kapitel 6 erläutert.

Die Bestimmung des anzufahrenden Zielpunktes unter Beachtung der Manövrierfähigkeit des Fahrzeuges wird im Kapitel 7 beschrieben.

Kapitel 8 behandelt die Regelungen zu Fahrzeugführung. Das Konzept zum rechnergestützten nichtlinearen Reglerentwurf wird im Detail vorgestellt.

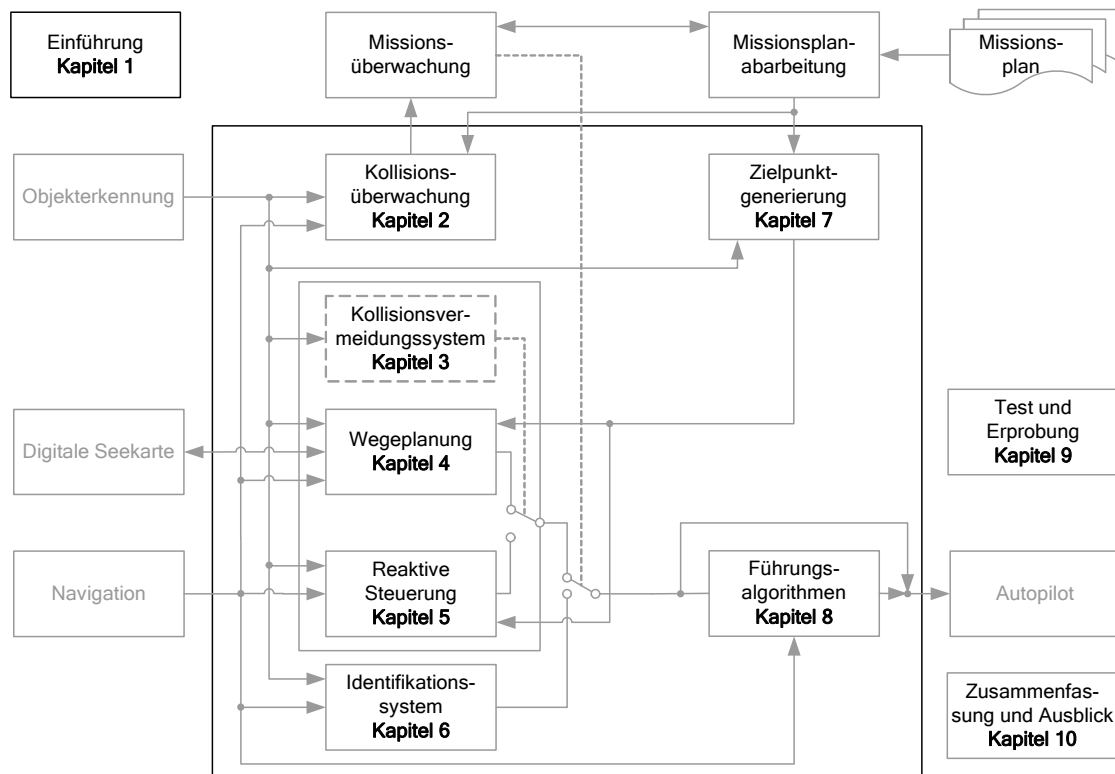


Abbildung 1.18: Aufbau der Arbeit

Kapitel 9 demonstriert das Modul *FIS* bei der simulativen und praktischen Erprobung. Im ersten Teil dieses Kapitels wird die unter MATLAB<sup>®</sup>/Simulink<sup>®</sup> entwickelte Simulationsumgebung vorgestellt. Für die Seeveruche wurde der Versuchsträger AUV M62 eingesetzt.

Abschließend erfolgt in Kapitel 10 eine Zusammenfassung der Arbeit. Hier werden die wesentlichen Neuheitsmerkmale dieser Arbeit noch einmal kompakt vorgestellt und mögliche Erweiterungen dargelegt.

Diese Arbeit enthält die folgenden neuartigen Ansätze:

- Ein mehrstufiges Verfahren zur effizienten Bestimmung einer Kollisionssituation zwischen elliptischen Zylindern und Strecken bzw. Kreisbögen.
- Die Entwicklung eines hybriden Kollisionsvermeidungssystems zur Gewährleistung des Echtzeitverhaltens des Systems.
- Ein neuartiges Verfahren zur Erzeugung von Gradientenlinien durch geometrische Konstruktion für ein reaktives Ausweichverfahren.
- Eine effiziente Quadtree-Zellteilung für einen Konfigurationsraum mit elliptischen Zylindern.
- Die Untersuchung mehrerer Ansätze zur Online-Erzeugung eines geometrischen Graphen sowie von Suchverfahren.
- Die Einbeziehung der Strömungsinformation bei der Wegeplanung.
- Der Einsatz von evolutionären Optimierungsverfahren und spezieller Gütekriterien zur Bestimmung der Reglerparameter unter Beachtung der Robustheitsanforderungen.
- Die Entwicklung einer Simulations- und Testumgebung unter MATLAB<sup>®</sup>/Simulink<sup>®</sup>.

## 2 Kollisionsüberwachung

Die Aufgabe des Submoduls *Kollisionsüberwachung* besteht in der Bestimmung der nächsten Kollisionssituation des Fahrzeuges mit einem Objekt auf der Basis der Fahrtroute und der detektierten Objekte. Die in diesem Zusammenhang interessierenden Parameter sind die Kollisionsposition und die Zeitdauer bis zur Kollision. Dazu überprüft die *Kollisionsüberwachung* eine Liste mit dem aktuellen Basismanöver und den zukünftigen Basismanövern auf eine mögliche Kollision mit allen detektierten Objekten.

In diesem Kapitel findet nach einer einführenden Vorstellung bekannter Ansätze und Einsatzfälle von Kollisionserkennungsverfahren in der Robotik eine Anforderungsbeschreibung für die *Kollisionsüberwachung* statt. Anschließend wird das entwickelte Konzept im Überblick vorgestellt. Dabei wird das mehrstufige Verfahren zur Kollisionserkennung mit seinen einzelnen Algorithmen detailliert beschrieben. Die Laufzeitüberprüfung des Verfahrens bildet den Abschluss dieses Kapitels.

### 2.1 Einführung in die Kollisionserkennung

Die Kollisionserkennung hat fundamentale Bedeutung in den Bereichen der Robotik (Kollisionsvermeidung, Bewegungsplanung), der Computerspiele (Navigation und Interaktion in 3D-Spielwelten), des CADs (*Computer Aided Design*) (Montageplanung und -simulation, Virtual Prototyping) sowie der Virtual Reality (Flugsimulatoren, Chirurgesimulationen, Telepräsenz). Sie umfasst die

*Untersuchung zweier Objekte auf Berührung bzw. Durchdringung.*

In [75, 151] wird für die Untersuchung von stationären Objekten der Begriff *statische Kollisionserkennung* verwendet. Diese Untersuchung kann im einfachsten Fall der Test zweier stationärer Objekte auf Überschneidung sein. Dies bezeichnet man auch als Kollisions-, Überlappungs- oder Durchdringungstest. Das Ergebnis eines solchen Tests ist die boolesche Aussage, ob sich die Objekte schneiden oder nicht. Für einige Anwendungen reicht diese Information aus (Virtual Prototyping, Computerspiele). Zusätzlich können bei der *statischen Kollisionserkennung* die Schnittpunkte und -kurven, die Überlappungsräume sowie die Objektabstände bestimmt werden. Findet eine Überprüfung von bewegten Objekten statt (*dynamische Kollisionserkennung*), interessiert neben der Aussage, ob es eine Kollision gab, auch der Zeitpunkt der ersten Berührung der beiden Objekte und die Kollisionsposition.

Die Grundanforderung an die Algorithmen zur Kollisionserkennung ist vielfach die Echtzeitfähigkeit bei einer großen Anzahl von zu untersuchenden Objekten, welche aus sehr komplexen geometrischen Formen bestehen können. So existiert eine Vielzahl von effizienten Methoden zur Kollisionserkennung auf Basis mathematischer Berechnungen, algorithmischer Geometrie und geometrischer Datenstrukturen.

### 2.1.1 Bekannte Ansätze

Für die Kollisionserkennung müssen die zu untersuchenden Objekte in einer mathematisch beschreibbaren Form vorliegen. Vielfach werden die Objekte durch konvexe Polyeder nachgebildet. Die Flächen dieser Polyeder bestehen aus  $n$ -Ecken (Polygone), wodurch eine sehr gute Nachbildung von beliebig komplexen Objektformen möglich ist. Deshalb werden sie für die Nachbildung von Lebewesen [104], Textilien [108] oder für Terrains [123] verwendet. Zur Abstandsbestimmung zwischen zwei konvexen Polyedern wird vielfach [76, 108] der GJK-Algorithmus [65] eingesetzt. Wenn die Objekte bzw. ihre Oberflächen durch mathematische Funktionen beschreibbar sind, können die Kollisionserkennungsalgorithmen direkt mit diesen Funktionen arbeiten. In [151] werden Verfahren zur Kollisionserkennung zwischen stationären (*statische Kollisionserkennung*) und sich bewegenden Körpern (*dynamische Kollisionserkennung*) vorgestellt, deren Oberflächen sich aus gekrümmten Flächen- und Kurvenstücken zusammensetzen. Als Flächen werden Ebenen, Quadriken (algebraische Flächen 2. Ordnung) und der Torus eingesetzt. Ein Problem bei diesen Verfahren besteht in der numerischen Robustheit der Berechnungen. Die Berechnungen für die *statische Kollisionserkennung* werden auf Polynome 4. Ordnung zurückgeführt, wodurch robuste und effiziente Verfahren zur Nullstellenbestimmung einsetzbar sind. Zur Senkung der Abarbeitungszeit bei der Kollisionserkennung haben sich Hüllkörper-Hierarchien (Bounding volume hierarchies BVHs) bewährt. Dabei wird das Objekt durch einfache geometrische Formen umschlossen, welche in einer hierarchischen Baumstruktur angeordnet sind. In der Wurzel des Baumes befindet sich ein Hüllkörper, der das gesamte Objekt umschließt. In seinen darunter liegenden Knoten sind kleinere Hüllkörper angeordnet, die Teile des Objektes einschließen. Mit zunehmender Tiefe findet so eine immer feinere Anpassung an die reale Objektform statt. Als Hüllkörper kommen Kugeln [87],  $r$ -Zylinder [76], achsengerichtete Boxen (AABB) [112] oder orientierte Boxen (OBB) [66, 123] zum Einsatz (siehe Abbildung 2.1). Bei der Kollisionserkennung zweier Objekte werden ihre Hüllkörper in den einzelnen Stufen gegeneinander durch einfache und schnelle Überlappungstests überprüft. Findet eine Überlappung statt, wird im darunter liegenden Teilzweig die Kollisionsüberprüfung fortgesetzt, bis keine Kollision mehr auftritt oder die tiefste Stufe (Blätter des Baumes) erreicht wurde. Für einige Anwendungen reicht die Kollisionsinformation der Hüllkörper aus [123]. Bei anderen Anwendungen schließen die Hüllkörper der tiefsten Stufe die eigentlichen Objektformen ein, so dass bei einer ermittelten Kollision zwischen den Hüllkörpern die aufwendigen Kollisionsberechnungen zwischen den exakten Objektformen durchgeführt werden [76].

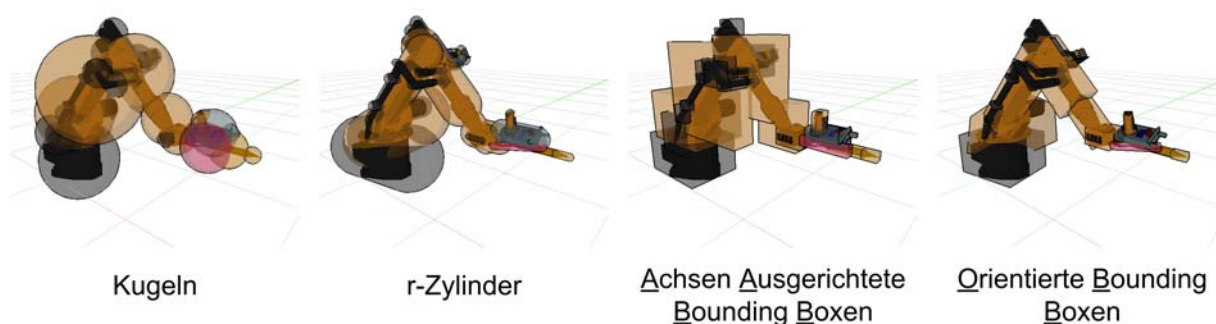


Abbildung 2.1: Hüllkörper [129]

Sind viele Objekte auf Kollision zu überprüfen, können *Sweep and Prune Algorithmen* eingesetzt werden [36, 112]. Dabei wird für jedes Objekt ein Hüllkörper in Form einer achsengerichteten Box definiert. Anschließend findet für jede Koordinatenachse ein aufsteigendes Sortieren der linken Intervallgrenzen der Boxen in einer Liste statt. Für die einzelnen Achsen werden dann diese Listen durchlaufen, um mögliche Überlappungen zwischen zwei Intervallen zu bestimmen. Gibt es zwischen zwei Boxen in jeder Achse eine Überlappung, kann eine exakte Kollisionserkennung zwischen ihren Objektformen durchgeführt werden. Dieses Vorgehen bezeichnet man auch als Zwei Phasen-Algorithmus [112].

### 2.1.2 Einsatzfälle bei autonomen mobilen Systemen und in der Robotik

Die Kollisionserkennung besitzt eine wichtige Bedeutung in der Robotik bei der Planung von kollisionsfreien Bewegungen. Ein schnelles Kollisionserkennungsverfahren zur Bewegungsplanung für den Roboter Puma 260 findet man in [76]. Die Modellierung des Roboters erfolgt dort durch Polyeder. Die Erkennung einer Kollisionssituation basiert auf Abstandsberechnungen zwischen den Objekten. Die Verwendung von statischen und dynamischen achsengerichteten Bounding Box (AABB) Hierarchien und einer parallelen Abarbeitung der Algorithmen führen zu einer Beschleunigung bei der Kollisionserkennung. Für einen autonomen Helikopter wird in [123, 124] ein Collision Checker zur Überprüfung der generierten Wege vorgestellt. Die einzelnen Polygone, welche die Umwelt mit ihren Gebäuden, Bäumen und Masten beschreiben, sind durch orientierte Boxen (OBB) in einem hierarchisch aufgebauten Baum (OBBTree) nachgebildet. Jede einzelne Box des Baumes ist um einen Sicherheitsbereich und den maximalen Radius des Helikopters vergrößert. Ein Polynom 3. Ordnung beschreibt die Flugbahn des Helikopters. Bei der Kollisionserkennung findet eine rekursive Schnittüberprüfung dieses Polynoms mit den einzelnen Boxen des Baumes, beginnend mit dem obersten Baumknoten, statt [124]. Dies ermöglicht eine effiziente Kollisionsüberprüfung unter Echtzeitbedingungen. Zur autonomen und automatischen Führung eines Schiffes auf Binnenwasserstraßen entwickelte die Universität Stuttgart ein integriertes Navigationssystem [15]. Dieses System soll die Sicherheit auf den Wasserstraßen erhöhen, den Schiffsführer von eintönigen Routineaufgaben befreien und ihn in schwierigen Situationen unterstützen. Um frühzeitig eine Konfliktsituation zu erkennen, werden die möglichen zukünftigen Fahrtrajektorien der fremden Schiffe ermittelt und zusammen mit der Leitlinie des eigenen Schiffes in einer Simulation analysiert. Dabei ist ein Sicherheitsbereich in Form eines Rechteckes um jedes Schiff definiert [14], der den Mindestabstand zwischen zwei Schiffen beschreiben soll. Auf der Basis der Fahrtrajektorien der Schiffe findet die Positionsbestimmung der Sicherheitsbereiche zu diskreten zukünftigen Zeitpunkten statt, welche dann auf Überschneidungen überprüft werden. Für den regelbasierten geometrischen Wegeplaner eines Reinigungsroboters wird eine kartengestützte Kollisionsanalyse zwischen dem geplanten Weg des Roboters und den bekannten Konturen des zu reinigenden Raumes durchgeführt [78]. Die einzelnen Konturen sind durch Streckenabschnitte beschrieben. Bei dieser Kollisionsanalyse wird ein geradliniger Bewegungskorridor auf Überschneidungen mit den Konturen überprüft. Findet eine Kollision statt, berechnet das Verfahren ein kollisionsfreies Wendemanöver unter Berücksichtigung der räumlichen Konturen sowie des Wenderadius und der Abmaße des Fahrzeuges. So entsteht ein mäanderförmiger Weg, welcher beim Abfahren zu einer flächendeckenden Reinigung des Raumes führt.

## 2.2 Konzept der Kollisionsüberwachung

### 2.2.1 Anforderungen

Die Abarbeitung der *Kollisionsüberwachung* erfolgt zyklisch in jeder Updateroutine des Moduls *FIS* alle 100 ms. Da die *Kollisionsüberwachung* mit den anderen Submodulen zusammen abgearbeitet wird, darf sie nur einen Teil der Rechenressourcen verwenden, um die Echtzeitforderung des Moduls *FIS* nicht zu verletzen. Die Anzahl der zu überprüfenden Objekte beträgt maximal 50. Dies resultiert aus dem vorgegebenen Sonarsektorbereich, der möglichen Objektausdehnung und dem vorhandenen Umweltszenario. Geht man von einer Fünf-Minuten-Vorhersage für die *Kollisionsüberwachung* aus, entspricht dieser Zeithorizont maximal 20 abzufahrenden Basismanövern. Die durchzuführende Kollisionserkennung ermittelt die Kollisionsposition und die Zeitdauer bis zur Kollision. Diese Werte werden an das Modul *Missionsüberwachung* zur Festlegung der nächsten Aktivitäten übergeben. Eine millimetergenaue Bestimmung der Kollisionsposition ist jedoch nicht erforderlich. Hier geht es vielmehr darum, eine mögliche Kollisionssituation sicher zu erkennen und das vorhandene Variieren der Objektparameter, das Abweichen von der vorgegebenen Sollbahn und die Manövrierfähigkeit des Fahrzeuges bei der Kollisionserkennung mit einzubeziehen.

### 2.2.2 Vorstellung des Verfahrens

Dieser Abschnitt stellt das Verfahren zur *Kollisionsüberwachung* im Überblick vor. Aufgabe ist es, den zukünftigen Fahrzeugtrajektorienverlauf auf eine Kollision mit den Objekten zu überprüfen. Dazu werden die  $m_{\text{Manöver}}$  Basismanöver mit den  $n_{\text{Objekt}}$  detektierten Objekten auf Kollision überprüft. Die Überprüfung erfolgt dabei in einem zweischleifigen Prozess (siehe Abbildung 2.2). Jedes einzelne Manöver wird nach der Reihenfolge seiner Abarbeitung im Missionsplan mit der gesamten Liste der Objekte übergeprüft. Es findet quasi ein simuliertes Abfahren der Mission statt. Gibt es eine Kollision zwischen dem Manöver und einem Objekt, wird die Zeit bis zur Kollision  $t_{\text{coll\_akt}}$  und die Kollisionsposition  $\mathbf{x}_{\text{coll}}$  abgespeichert. Tritt eine weitere Kollisionssituation dieses Manövers mit einem Objekt aus der Objektliste auf, werden die Kollisionsparameter übernommen, wenn die Kollisionszeit kürzer als die momentan abgespeicherte Zeit ist. Wenn bei der Überprüfung eines Manövers mit der gesamten Objektliste eine Kollision auftritt, brauchen die restlichen Manöverelemente nicht mehr überprüft zu werden. Gab es keine Kollision zwischen dem überprüften Manöver und der Objektliste, wird die Zeit zum Abfahren des Manövers  $t_{\text{Man}_i}$  als aktuelle Kollisionszeit  $t_{\text{coll\_akt}}$  übernommen und der Kollisionszeit  $t_{\text{coll}}$  hinzugefügt. Solange keine Kollision auftritt, enthält  $t_{\text{coll}}$  die Zeit zum Abfahren der  $i$  geprüften Manöver. Die Anzahl der durchzuführenden Kollisionstests  $num_{\text{Test}}$  beträgt so im ungünstigsten Fall (d.h. es gab keine Kollision zwischen den einzelnen Manövern und der Objektliste):

$$num_{\text{Test}} = m_{\text{Manöver}} \cdot n_{\text{Objekte}} \quad (2.1)$$

Dies entspricht bei den Vorgaben des Abschnitts 2.2.1 bei 20 Manövern und 50 Objekten 1000 Kollisionstests. Im nachfolgenden Abschnitt wird das Verfahren zur Kollisionserkennung zwischen den einzelnen Manövern und den Objekten im Detail vorgestellt. Bei der Entwicklung des Verfahrens stand neben der sicheren und robusten Berechnung auch die Forderung nach einer schnellen Lieferung der Kollisionsanalyse des Operationsgebietes.

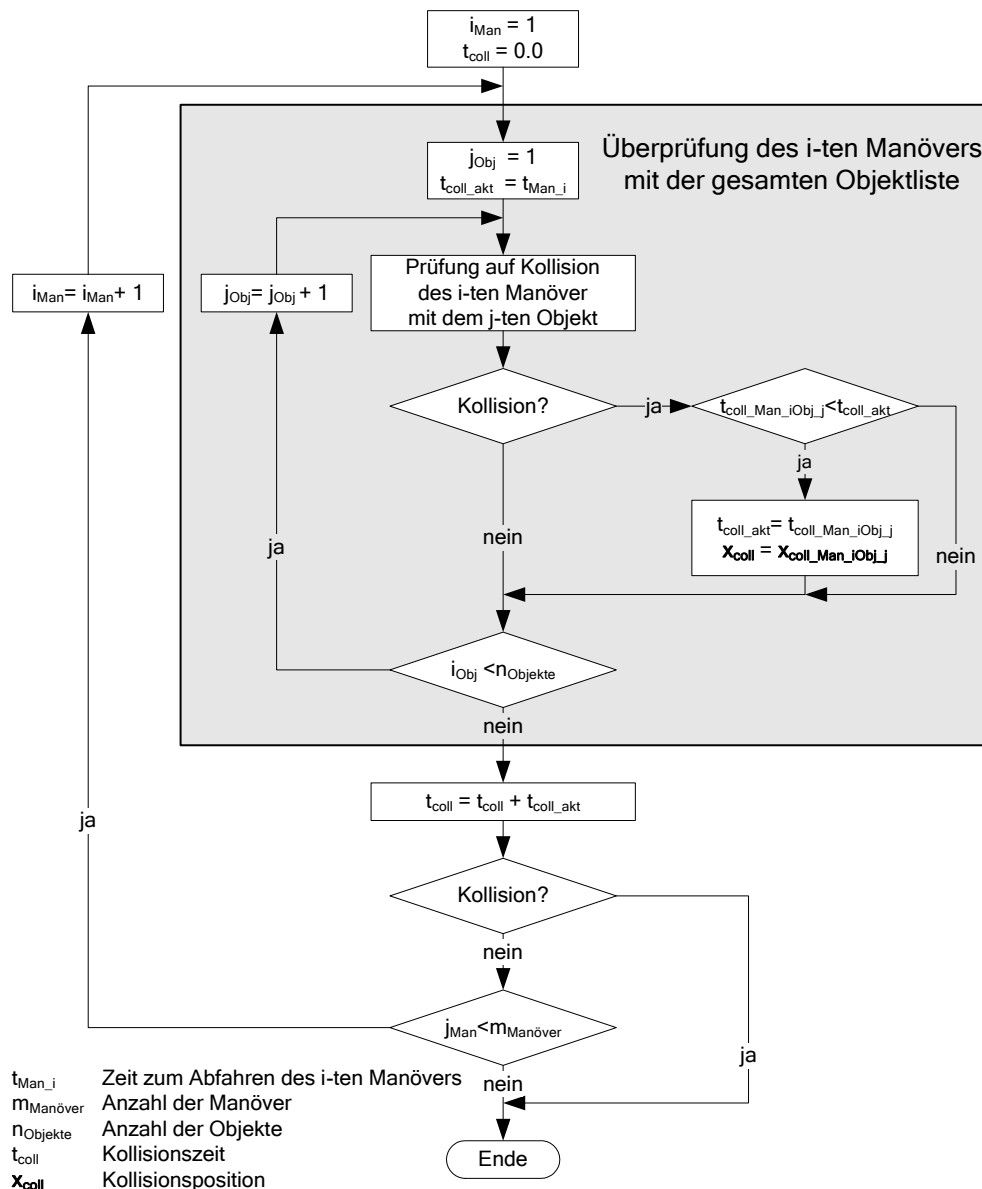


Abbildung 2.2: Flussdiagramm zur Bestimmung der ersten Kollisionssituation

## 2.3 Kollisionserkennung

### 2.3.1 Problemstellung

Bei der Ermittlung einer Kollisionssituation zwischen dem Fahrzeug und den Objekten handelt es sich um ein Problem der *dynamischen Kollisionserkennung*. Dabei ist das Fahrzeug das bewegte Objekt, welches auf eine Kollision mit den detektierten stationären Objekten überprüft wird. Bei einer exakten Kollisionsermittlung müsste das Fahrzeug, mit seiner komplexen geometrischen Form und seinem durch das Manöver beschriebenen Bewegungsverlauf, auf Durchdringung mit den elliptischen Objektzylindern überprüft werden. Dies wäre sehr rechenaufwendig und durch die nicht exakte Vorhersage der real gefahrenen Trajektorie auch nicht möglich. Aus diesem Grund findet eine Approximation des Modells zur Kollisionserkennung statt (siehe Abbildung 2.3). Dabei wird das Fahrzeug als punktförmiger Körper modelliert.

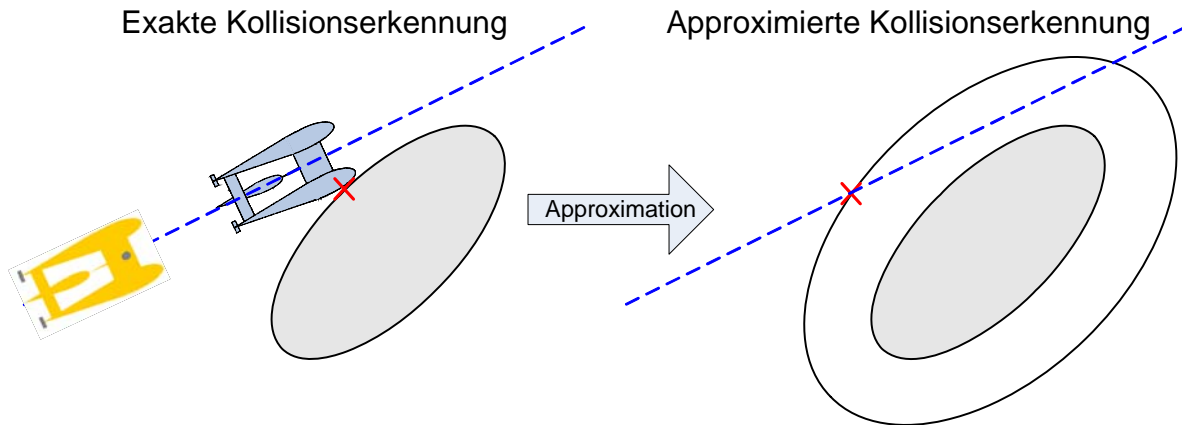


Abbildung 2.3: Approximation der Kollisionserkennung

Die Objekte werden durch die Vergrößerung der Haupt- und Nebenachse, der elliptischen Grundfläche und der Höhe modifiziert. Der Wert der Vergrößerung berücksichtigt einen Sicherheitsabstand, die Fahrzeugabmaße, mögliche Abweichungen des Fahrzeuges von der vorgegebenen Sollbahn und die Manövrierfähigkeit des Fahrzeuges. Durch diese Approximation können die Algorithmen zur Kollisionserkennung auf Schnittberechnungen zwischen den Kurvenverläufen der Sollbahn und einem elliptischen Zylinder zurückgeführt werden.

Eine weitere Approximation im Kollisionsmodell wäre die Nachbildung der Objektoberflächen durch Polygone. Dies führt bei stark gekrümmten Oberflächen zu einer Vielzahl von Polygonen, was im Aufbau der Objektoberflächen und bei der Kollisionserkennung sehr rechenintensiv sein kann. Deshalb sollte dieses Verfahren nicht verwendet werden, wenn die gekrümmten Objektoberflächen durch eine mathematische Funktion beschreibbar sind und so die Algorithmen direkt mit den Oberflächen arbeiten können, ohne sie durch Polygone nachzubilden [151]. Dies ist bei der Form des elliptischen Zylinders und den Kurvenverläufen gegeben und soll im nachfolgenden Abschnitt vorgestellt werden.

## 2.3.2 Bestimmung der Kollisionsparameter

In diesem Abschnitt werden die Algorithmen zur Ermittlung der Kollisionsposition und der Kollisionszeit vorgestellt. Da die Manöver nur aus Kreisbögen und Strecken aufgebaut sind, beschränken sich die dreidimensionalen Schnittberechnungen zur Ermittlung der Kollisionsposition auf:

- Schnitt Strecke – Elliptischer Zylinder
- Schnitt Kreisbogen – Elliptischer Zylinder.

### 2.3.2.1 Strecke – Elliptischer Zylinder

Durch die achsenparallele z-Ausrichtung des elliptischen Zylinders erfolgt der erste Teil der Schnittberechnung mit einer auf der x-y Ebene projizierten Strecke und Ellipse.

Eine Strecke wird in Parameterdarstellung durch die Beziehung

$$\mathbf{x} = \mathbf{a} + t\mathbf{v} \quad t \in [0,1] \quad (2.2)$$

beschrieben.



Die einzelnen Koordinaten berechnen sich zu:

$$\begin{aligned} x &= x_a + tx_v \\ y &= y_a + ty_v \end{aligned} \quad (2.3)$$

Die quadratische Gleichung zur Beschreibung einer Ellipse lautet:

$$ax^2 + bxy + cy^2 + dx + ey + f = 0. \quad (2.4)$$

Die Bestimmung der einzelnen Parameter  $a$  bis  $f$  aus den Kenngrößen der Ellipse wird im Anhang dieser Arbeit detailliert beschrieben. Für die Schnittberechnung wird die Ellipse in den Koordinatenursprung transformiert. Dies führt zu einer schnelleren und vor allem robusteren Lösung. Durch diese Transformation werden die Mittelpunktkoordinaten der Ellipse zu Null, so dass die Parameter  $d$  und  $e$  wegfallen und der Parameter  $f$  zu  $f_{center}$  modifiziert wird:

$$ax^2 + bxy + cy^2 + f_{center} = 0. \quad (2.5)$$

Die Ursprungsposition  $\mathbf{a}$  der Strecke verschiebt sich dabei um den Ortsvektor des Ellipsenmittelpunktes  $\mathbf{x}_{Ellipse}$  zu

$$\mathbf{a}_{trans} = \mathbf{a} - \mathbf{x}_{Ellipse}. \quad (2.6)$$

Die Gleichung (2.3) mit dem transformierten Ursprungspunkt  $\mathbf{a}_{trans}$  wird in Gleichung (2.5) eingesetzt:

$$a(x_{a\_trans} + tx_v)^2 + b(x_{a\_trans} + tx_v)(y_{a\_trans} + ty_v) + c(y_{a\_trans} + ty_v)^2 + f_{center} = 0 \quad (2.7)$$

wodurch eine quadratische Gleichung  $c_2 t^2 + c_1 t + c_0 = 0$  mit den Parametern:

$$\begin{aligned} c_0 &= ax_{a\_trans}^2 + bx_{a\_trans}y_{a\_trans} + cy_{a\_trans}^2 + f_{center} \\ c_1 &= b(x_v y_{a\_trans} + x_{a\_trans} y_v) + 2(ax_{a\_trans}x_v + cy_{a\_trans}y_v) \\ c_2 &= ax_v^2 + bx_v y_v + cy_v^2 \end{aligned} \quad (2.8)$$

entsteht. Gibt es für diese quadratische Gleichung reelle Nullstellen  $t_1$  und  $t_2$  und liegen diese im Intervall  $[0,1]$ , wird der kleinere Wert der beiden als Variable  $t$  in die dreidimensionale Parameterdarstellung der Strecke:

$$\begin{aligned} x_{coll} &= x_a + tx_v \\ y_{coll} &= y_a + ty_v \\ z_{coll} &= z_a + tz_v \end{aligned} \quad (2.9)$$

eingesetzt. Befindet sich der Schnittpunkt  $\mathbf{x}_{coll}$  oberhalb oder unterhalb des Zylinders, oder liegt die Ursprungsposition der Strecke  $\mathbf{a}$  direkt über oder unter dem Zylinder, wird in Abhängigkeit dieser Lage eine Schnittüberprüfung der Strecke mit der oberen bzw. unteren Abdeckung des elliptischen Zylinders durchgeführt. Es findet eine Schnittberechnung zwischen einer Strecke und einer Ebene statt. Die Variable  $t$  der Strecke wird durch die Beziehung:

$$t = \frac{z_{Ebene} - z_a}{z_v} \quad (2.10)$$

bestimmt. Die Kollisionsposition wird nach Gleichung (2.9) bestimmt. Abschließend wird überprüft, ob sich der Schnittpunkt innerhalb der elliptischen Abdeckung befindet. Dazu werden die berechneten Koordinaten  $x_{coll}$  und  $y_{coll}$  in Gleichung (2.4) eingesetzt. Ist das Ergebnis negativ, liegt die Kollisionsposition innerhalb der elliptischen Abdeckung. Die Kollisionszeit bestimmt sich aus dem bis zur Kollisionsposition zurückgelegten Streckenabschnitt  $tv$  und der vorgegebenen Fahrtgeschwindigkeit  $v_{Bahn}$  zu:

$$t_{coll} = \frac{\|tv\|}{v_{Bahn}}. \quad (2.11)$$

### 2.3.2.2 Kreisbogen – Elliptischer Zylinder

Die Schnittberechnungen gehen in Analogie zum Abschnitt 2.3.2.1 von einer zweidimensionalen Betrachtung aus. Dabei wird der Kreisbogen als Kreis mit der quadratischen Gleichung:

$$x^2 + y^2 + d_{Kreis}x + e_{Kreis}y + f_{Kreis} = 0 \quad (2.12)$$

beschrieben. Die Ellipse wird in den Koordinatenursprung transformiert, wodurch ihre Gleichung zu:

$$a_{Ellipse}x^2 + b_{Ellipse}xy + c_{Ellipse}y^2 + f_{Ellipse\_center} = 0 \quad (2.13)$$

wird. Die Parameter der Kreisgleichung werden durch Verschiebung des Kreismittelpunktes um den Ortsvektor des Ellipsenmittelpunktes  $\mathbf{c}_{Kreis\_trans} = \mathbf{c}_{Kreis} - \mathbf{c}_{Ellipse}$  zu

$$\begin{aligned} d_{Kreis} &= -2.0x_{c\_Kreis\_trans} \\ e_{Kreis} &= -2.0y_{c\_Kreis\_trans} \\ f_{Kreis} &= f_{Kreis\_center} + x_{c\_Kreis\_trans}^2 + y_{c\_Kreis\_trans}^2 \end{aligned} \quad (2.14)$$

modifiziert. Diese Transformation bewirkt, dass die Parameter  $d$ ,  $e$  und  $f$  der Gleichungen (2.12) und (2.13) kleiner bzw. zu Null werden, was die numerische Stabilität des zu lösenden Gleichungssystems erhöht. Bei diesem Gleichungssystem handelt es sich um ein System aus zwei Variablen und zwei quadratischen Gleichungen. Die Lösung eines solchen Gleichungssystems wird im Anhang detailliert beschrieben. Als Ergebnis erhält man für die y-Koordinate der einzelnen Schnittpunkte ein Polynom 4. Ordnung:

$$h_4y^4 + h_3y^3 + h_2y^2 + h_1y + h_0 = 0 \quad (2.15)$$

mit den Parametern:

$$\begin{aligned} h_0 &= d_{00}d_{10} - d_{20}^2 \\ h_1 &= d_{01}d_{10} + d_{00}d_{11} - 2d_{20}d_{21} \\ h_2 &= d_{01}d_{11} + d_{00}d_{12} - d_{21}^2 - 2d_{20}d_{22} \\ h_3 &= d_{01}d_{12} + d_{00}d_{13} - 2d_{21}d_{22} \\ h_4 &= d_{01}d_{13} - d_{22}^2 \end{aligned} \quad (2.16)$$

Die einzelnen Variablen  $d_{xx}$  werden durch einen Koeffizientenvergleich der Gleichungen (2.12) und (2.13) mit den in Abschnitt A.1.2 beschriebenen vollständig besetzten quadratischen Gleichungen zu:

$$\begin{aligned}
d_{00} &= a_{\text{Ellipse}} d_{\text{Kreis}} \\
d_{01} &= -b_{\text{Ellipse}} \\
d_{10} &= -d_{\text{Kreis}} f_{\text{Ellipse}} \\
d_{11} &= b_{\text{Ellipse}} f_{\text{Kreis}} \\
d_{12} &= b_{\text{Ellipse}} e_{\text{Kreis}} - d_{\text{Kreis}} c_{\text{Ellipse}} \\
d_{13} &= b_{\text{Ellipse}} \\
d_{20} &= a_{\text{Ellipse}} f_{\text{Kreis}} - f_{\text{Ellipse\_center}} \\
d_{21} &= a_{\text{Ellipse}} e_{\text{Kreis}} \\
d_{22} &= a_{\text{Ellipse}} - c_{\text{Ellipse}}
\end{aligned} \tag{2.17}$$

bestimmt. Die Nullstellen des Polynoms 4. Ordnung der Gleichung (2.15) bestimmen sich nach der Formel von Ferrari [153].

Jede reelle Nullstelle von Gleichung (2.15) ist eine y-Koordinate eines Schnittpunktes. Die dazugehörige x-Koordinate erhält man durch Einsetzen der einzelnen Nullstellen in Gleichung (2.12). Um die dabei gefundenen Scheinlösungen auszuschließen, werden die ermittelten Koordinatenpaare in Gleichungen (2.12) und (2.13) eingesetzt. Wenn beide Gleichungen als Ergebnis den Wert Null besitzen (mit vorgegebenen Toleranzen), ist die ermittelte Position ein Schnittpunkt. Abschließend wird für jeden Schnittpunkt der Winkelwert vom Kreiszentrum aus bestimmt. So können die Schnittpunkte, welche außerhalb des Kreisbogens liegen, ausgeschlossen und der erste Schnittpunkt auf dem Kreisbogen bestimmt werden. Die Bestimmung der Kollisionszeit erfolgt in Analogie zum Abschnitt 2.3.2.1.

### 2.3.3 Das Zwei-Phasen-Konzept zur Kollisionserkennung

Durch die mathematisch aufwendigen Schnittberechnungen kann nach Gleichung (2.1) die Gesamtrechnenzeit zur Ermittlung einer Kollisionssituation bei einer großen Manöver- und/oder Objektanzahl sehr hoch sein. Aus diesem Grund und durch die geringe Wahrscheinlichkeit von Überschneidungen zwischen den Manövern und Objekten innerhalb des Operationsgebietes wurde ein mehrstufiges Konzept zur Bestimmung einer Kollisionssituation entworfen. Die Idee basiert auf dem bei Kollisionserkennungsverfahren oft eingesetzten Zwei-Phasen-Algorithmus. In der *Weiten Phase* werden durch einfache Tests alle Paare ausgeschlossen, welche sich nicht überlappen. In der *Nahen Phase* findet dann die eigentliche Kollisionserkennung mit den übrig gebliebenen Paaren statt.

Die Manöver und Objekte werden in der *Weiten Phase* durch einfache geometrische Formen (Hüllkörper) nachgebildet. Dabei schließt der Hüllkörper das Manöver bzw. das Objekt vollkommen ein, unter der Bedingung, dass sein Volumen möglichst gering ist. In der ersten Stufe der *Weiten Phase* werden alle Manöver und Objekte durch achsengerichtete Boxen nachgebildet und auf Überlappung überprüft. Die Vorstellung des Verfahrens erfolgt im Abschnitt 2.3.4. Kommt es bei dieser Stufe zu einer Überlappung, wird in einer zweiten Stufe ein Überlappungstest zwischen zwei orientierten Boxen oder zwischen einer orientierten Box und einer Strecke durchgeführt (siehe Abbildung 2.4). Eine Beschreibung dieser Verfahren befindet sich in Abschnitt 2.3.5.

	Geometrische Form	Weite Phase		Nahe Phase
		1. Stufe Achsenorientierte Boxen (AABB)	2. Stufe Orientierte Boxen (OBB)	Exakte Formen
Objekt	Elliptischer Zylinder			
	Strecke			
	Kreisbogen			
Manöver	Kreis			
		<div style="display: flex; align-items: center; justify-content: center;"> <div style="width: 100px; height: 100px; border: 1px solid black; margin-right: 10px;"></div> <div style="text-align: center;"> <p style="margin: 0;">Anzahl der Schnitttests, Volumen der Objekte, Approximationsfehler</p> <p style="margin: 0;">←</p> </div> </div> <div style="display: flex; align-items: center; justify-content: center; margin-top: 10px;"> <div style="width: 100%; height: 20px; background-color: #e0e0e0; border: 1px solid black; position: relative;"> <div style="position: absolute; right: -10px; top: -10px; width: 20px; height: 20px; border: 1px solid black; background-color: white;"></div> </div> <div style="text-align: center; margin-left: 10px;"> <p style="margin: 0;">→</p> <p style="margin: 0;">Rechenaufwand</p> </div> </div>		

Abbildung 2.4 Zwei Phasen-Konzept der Kollisionserkennung

Erst wenn auch dieser zweite Test erfolgreich war, werden die Algorithmen zur Bestimmung der Kollisionsparameter des Abschnittes 2.3.2 abgearbeitet. Die Zeit für den Test eines Manövers mit der gesamten Objektliste  $t_{\Sigma}$  beträgt bei diesem Zwei-Phasen-Konzept:

$$t_{\Sigma} = n_{WP1} t_{WP1} + n_{WP2} t_{WP2} + n_{NP} t_{NP}. \quad (2.18)$$

Dabei entspricht  $n_{WP1}$ ,  $n_{WP2}$  und  $n_{NP}$  der Anzahl der durchzuführenden Schnitttests in den einzelnen Stufen bzw. Phasen ( $WP1$ : erste Stufe der *Weiten Phase*,  $WP2$ : zweite Stufe der *Weiten Phase*,  $NP$ : *Nahe Phase*). Die Werte  $t_{WP1}$ ,  $t_{WP2}$  und  $t_{NP}$  sind die benötigten Abarbeitungszeiten der Tests. Eine Aussage über die durchschnittliche Zeit des Tests  $t_{\Sigma avg}$  kann unter Verwendung der Wahrscheinlichkeitsrechnung erfolgen. Dabei wird die Schnittwahrscheinlichkeit  $P(S)$  eingeführt, welche die Wahrscheinlichkeit einer Überlappung bzw. Überschneidung für die einzelnen Schnitttests beschreibt. Die Formel zur Ermittlung der durchschnittlichen Rechenzeit lautet unter Verwendung von Gleichung (2.18) und der Tatsache, dass in der ersten Stufe der *Weiten Phase* alle Objekte mit dem Manöver überprüft werden ( $n_{WP1} = n_{Objekte}$ ):

$$t_{\Sigma avg} = n_{Objekte} (t_{WP1} + P(S_{WP1}) t_{WP2} + P(S_{WP1}) P(S_{WP2}) t_{NP}). \quad (2.19)$$

Die Schnittwahrscheinlichkeit  $P(S)$  für die einzelnen Tests kann durch die Geometrische Wahrscheinlichkeit [106] oder durch praktische Versuche [17] auf Basis einer Monte-Carlo-Simulation ermittelt werden. Sie ist abhängig von den geometrischen Abmaßen des Manövers, des Objektes und des Operationsgebietes. Die Gleichung (2.19) liefert ein wichtiges Instrument zur Auswertung und zur Optimierung des Konzeptes der Kollisionserkennung. So kann basierend auf den Abarbeitungszeiten der Tests überprüft werden, ob die zweite Stufe der *Weiten Phase* eine Berechtigung besitzt und zur Beschleunigung des Verfahrens beiträgt. Wird nur mit der ersten Stufe der *Weiten Phase* gearbeitet, bestimmt sich die durchschnittliche Rechenzeit zu:

$$t_{\Sigma avg} = n_{\text{Objekte}} (t_{WP1} + P(S_{WP1})t_{NP}). \quad (2.20)$$

Eine Senkung der Abarbeitungszeit durch den Einsatz der zweiten Stufe der *Weiten Phase* liegt dann vor, wenn:

$$P(S_{WP1})t_{NP} > P(S_{WP1})t_{WP2} + P(S_{WP1})P(S_{WP2})t_{NP} \quad (2.21)$$

bzw.

$$\frac{t_{NP}}{t_{WP2}} > \frac{1}{1 - P(S_{WP2})} \quad (2.22)$$

ist. Das dazugehörige Effizienzdiagramm ist in Abbildung 2.5 dargestellt.

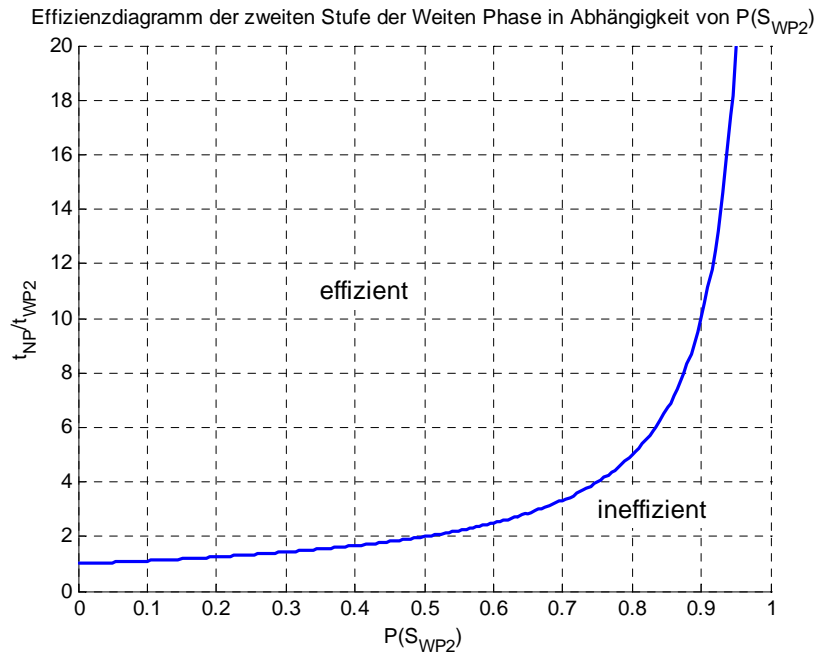


Abbildung 2.5: Effizienzdiagramm  $t_{NP}/t_{WP2} = f(P(S_{WP2}))$

### 2.3.4 Erste Stufe der Weiten Phase

In der ersten Stufe der *Weiten Phase* wird eine Begrenzungsbox in Form eines achsengerichteten Quaders (axis-aligned bounding box AABB) für das Manöver und das Objekt definiert [112, 135]. Der Algorithmus zur Überprüfung einer Überlappung zwischen zwei achsengerichteten Begrenzungsboxen wird in Abschnitt 2.3.4.1 vorgestellt.

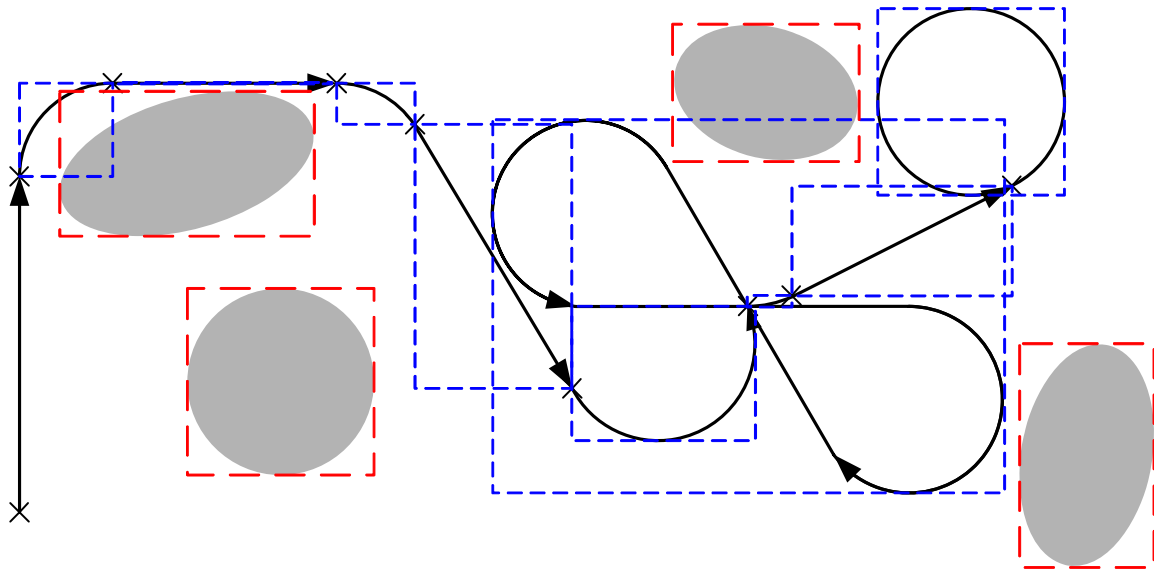


Abbildung 2.6: Beschreibung der Manöver und Objekte durch achsenparallele Rechtecke

Durch diesen Test kann die mögliche Anzahl von Überschneidungen zwischen Manövern und Objekten sehr eingeschränkt werden. In Abbildung 2.6 sind diese Begrenzungsboxen für die einzelnen Manöver und Objekte eines möglichen Missionsszenarios dargestellt. Dabei wurde ein Missionsplan aus 10 Basismanövern definiert, den das Fahrzeug abfahren soll. In diesem abzufahrenden Parcours befinden sich 4 Objekte. Gestartet wird im unteren, linken Teil mit einem Streckenmanöver. Beim zweiten Manöver (Kreisbogen) gibt es eine Überschneidung zwischen der Begrenzungsbox des Objektes und des Manövers. Für diese Kombination werden die weiteren Tests entsprechend Abbildung 2.4 durchgeführt. Die maximale Anzahl der bei diesem Missionsszenario durchzuführenden Tests beträgt bei den 10 Basismanövern und den 4 Objekten 40. Für das in Abbildung 2.6 dargestellte Szenario wurden zwei mögliche Überschneidungssituationen ermittelt. Nur für diese zwei Überschneidungssituationen brauchen die weiteren Tests durchgeführt zu werden.

### 2.3.4.1 Überlappungstest zwischen achsengerichteten Boxen

Die achsengerichtete Begrenzungsbox  $i$  wird durch die Punkte  $\mathbf{p}_{min}^i = [x_{min}, y_{min}, z_{min}]$  und  $\mathbf{p}_{max}^i = [x_{max}, y_{max}, z_{max}]$  beschrieben (siehe Abbildung 2.7). Diese Punkte charakterisieren die Ausdehnung der Box durch ihre minimale und maximale Begrenzung in den einzelnen Koordinatenachsen.

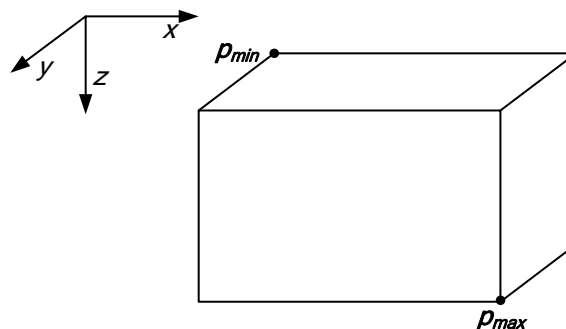


Abbildung 2.7: Definition der achsengerichteten Begrenzungsbox

Eine Überprüfung erfolgt für die einzelnen Koordinaten durch den gegenseitigen Größenvergleich der maximalen Begrenzung der einen mit der minimalen Begrenzung der zweiten Box. Findet bei einem dieser Tests keine Überlappung statt, so gibt es keine Überlappung der Boxen. Eine Überlappung zweier Boxen existiert nur, wenn alle Tests erfolgreich waren. Da bei diesem ersten Test nur Vergleichsoperationen durchgeführt werden, ist der benötigte Rechenaufwand sehr gering. Der Pseudocode ist in Abbildung 2.8 dargestellt.

```

bool SchnittAABBAABB3D(Box A, Box B)
{
    if  $p_{max}^A(x) < p_{min}^B(x)$  return false;
    if  $p_{max}^B(x) < p_{min}^A(x)$  return false;
    if  $p_{max}^A(y) < p_{min}^B(y)$  return false;
    if  $p_{max}^B(y) < p_{min}^A(y)$  return false;
    if  $p_{max}^A(z) < p_{min}^B(z)$  return false;
    if  $p_{max}^B(z) < p_{min}^A(z)$  return false;
    return true;
}

```

Abbildung 2.8: Pseudocode Überlappungstest von achsengerichteten Boxen

### 2.3.5 Zweite Stufe der Weiten Phase

Bei der zweiten Teststufe der *Weiten Phase* werden die Begrenzungsboxen für die Manöver *Kreisbogen*, *Kreis* und *Kreuzspirale* und das Objekt in der horizontalen Ebene so ausgerichtet, dass sie die kleinste Grundfläche bzw. das kleinste Volumen besitzen und das Manöver bzw. das Objekt vollständig einschließen. Die so entstandenen orientierten Boxen haben eine achsenparallele Ausrichtung in der z-Koordinate und besitzen nur eine Rotation um die z-Achse. Dies ist durch die gewählte Objektform und die Manöververläufe gegeben und führt zu einfacheren Berechnungen als bei beliebig orientierten Boxen, welche eine Rotation um alle drei Achsen besitzen können. Der durchzuführende Überlappungstest zwischen zwei orientierten Boxen wird im Abschnitt 2.3.5.1 vorgestellt.

Das Basismanöver *Strecke* wird nicht durch eine Begrenzungsbox umhüllt. Hier findet eine Überprüfung direkt mit seiner Geometrie und der zur Nachbildung des Objektes definierten orientierten Box statt. Die Schnittüberprüfung für eine Strecke und eine orientierte Box beschreibt Abschnitt 2.3.5.2.

In Abbildung 2.9 sind die Begrenzungsboxen der einzelnen noch zu untersuchenden Manöver und Objekte für das im vorhergehenden Abschnitt vorgestellte Missionsszenario dargestellt. Für die Tests der zweiten Stufe der *Weiten Phase* wurde eine Überschneidungssituation ermittelt. So konnte durch die im Vorfeld durchgeführten, vereinfachten Schnittberechnungen mit den Begrenzungsboxen, der Berechnungsaufwand von vierzig auf eine exakte Schnittberechnung gesenkt werden.

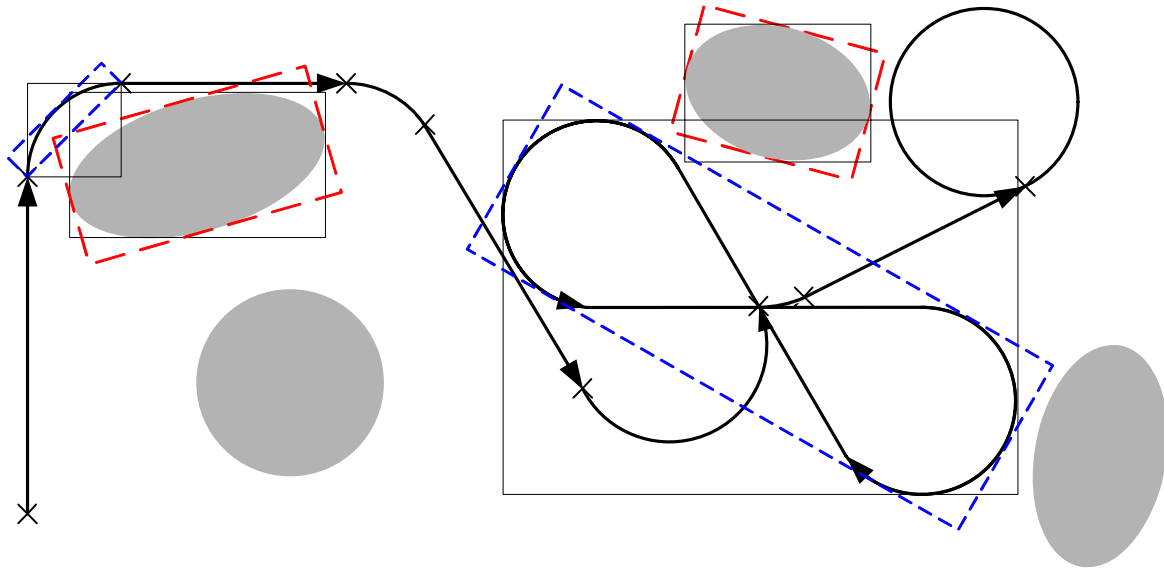


Abbildung 2.9: Beschreibung der Manöver und Objekte durch orientierte Begrenzungsboxen

### 2.3.5.1 Überlappungstest zwischen orientierten Boxen

Eine effiziente Methode zur Ermittlung einer Überlappungssituation bei beliebig orientierten Boxen wird in [66] vorgestellt. Nach [44] ist dieser Test deutlich schneller als bei einer Beschreibung der beiden Boxen durch Polyeder, wobei ihre einzelnen Flächen (Polygone) auf Überschneidung überprüft werden. Die Idee basiert auf dem *Separierenden Achsentheorem* (Separating Axis Theorem), welches lautet:

*Zwei konvexe Objekte schneiden sich nicht, wenn es eine Achse gibt, auf welcher sich ihre projizierten Intervalle nicht überlappen.*

Bei zwei beliebig im Raum orientierten Boxen ergibt sich die Anzahl der zu testenden Achsen aus den 3 Flächenrichtungen für jede der beiden Boxen und den 9 Kombinationen ihrer Kantenrichtungen zu 15. Eine detaillierte mathematische Beschreibung dieser Tests befindet sich in [44, 66] und [135]. Durch die Verwendung von Begrenzungsboxen mit einer achsenparallelen Ausrichtung in der  $z$ -Koordinate vereinfacht sich der Überlappungstest auf eine zweidimensionale Überprüfung der Grundflächen der Box und einen Größenvergleich ihrer Ausdehnungen in der  $z$ -Richtung, analog zu den achsengerichteten Boxen. Da der Überlappungstest zwischen zwei orientierten Boxen erst nach dem erfolgreichen Test der achsengerichteten Boxen durchgeführt wird, erfolgt schon dort ein Überlappungstest in der  $z$ -Richtung. So beschränkt sich der Überlappungstest auf zwei beliebig orientierte Rechtecke im 2D-Raum. Hierfür werden lediglich 4 Tests durchgeführt, die sich aus den zwei Kantenrichtungen der beiden Rechtecke ergeben. Nachfolgend soll dieses Verfahren detailliert vorgestellt werden.

Ein Rechteck  $i$  ist durch einen Zentrumsmittelpunkt  $c_i$ , die halben Längen seiner Abmaße  $a_{i1}$  und  $a_{i2}$  und die körperfesten Einheitsvektoren  $e_{i1}$  und  $e_{i2}$  zur Beschreibung seiner Orthonormalbasis definiert. Diese Vektoren entsprechen auch den Kantenrichtungen des Rechteckes. Bei den durchzuführenden Separationstests werden die auf die zu überprüfenden Achsen mit dem Einheitsvektor  $l_k$  ( $k = 1, 2, 3, 4$ ) projizierten Intervalle der Rechtecke A und B auf eine Trennung untersucht. Diese ist dann gegeben, wenn die Summe der Radien (halbe Intervalllänge)  $r_A$  und  $r_B$  kleiner als die Länge  $d$  ist. Damit gilt:



$$d > r_A + r_B. \quad (2.23)$$

Die Länge  $d$  beschreibt den Abstand zwischen den projizierten Mittelpunkten  $c_{A1}$  und  $c_{B1}$  der beiden Boxen. Bestimmt wird sie durch die Projektion des Abstandsvektors  $\mathbf{t}$  auf die zu überprüfende Achse  $\mathbf{l}_k$  nach der folgenden Gleichung:

$$d = |\mathbf{t} \cdot \mathbf{l}_k|. \quad (2.24)$$

Die Radien  $r_A$  und  $r_B$  bestimmen sich aus der Summe der Projektionen ihrer mit den halben Längen skalierten Einheitsvektoren auf die zu überprüfende Achse  $\mathbf{l}_k$  zu:

$$r_i = \sum_{j=1}^2 a_{ij} |\mathbf{e}_{ij} \cdot \mathbf{l}_k|. \quad (2.25)$$

Abbildung 2.10 zeigt den Überlappungstest für die Achse  $\mathbf{e}_{A1}$ . Die Gleichungen zur Berechnung der Werte  $r_A$ ,  $r_B$  und  $d$  für die einzelnen Tests sind in Tabelle 2.1 enthalten.

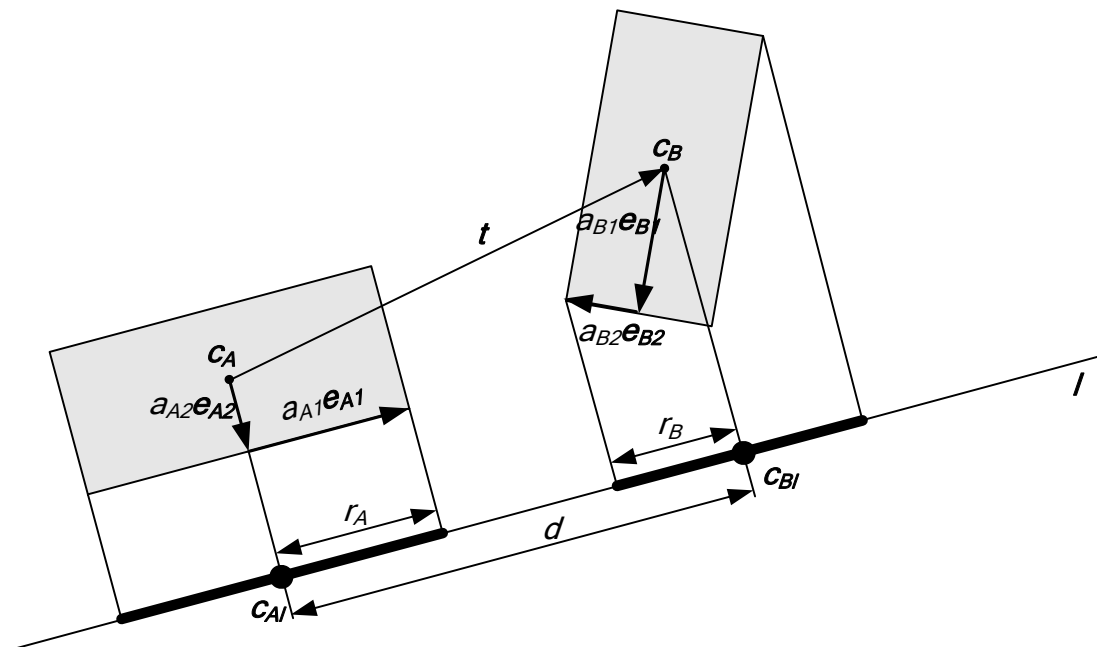


Abbildung 2.10: Überlappungstest durch Achsensparierung

Tabelle 2.1: Bestimmung der Radien  $r_A$  und  $r_B$  und des projizierten Boxenabstandes  $d$

Projektionsachse $\mathbf{l}_k$	$r_A$	$r_B$	$d$
$\mathbf{e}_{A1}$	$a_{A1}$	$a_{B1} \mathbf{e}_{B1} \cdot \mathbf{e}_{A1}  + a_{B2} \mathbf{e}_{B2} \cdot \mathbf{e}_{A1} $	$ \mathbf{t} \cdot \mathbf{e}_{A1} $
$\mathbf{e}_{A2}$	$a_{A2}$	$a_{B1} \mathbf{e}_{B1} \cdot \mathbf{e}_{A2}  + a_{B2} \mathbf{e}_{B2} \cdot \mathbf{e}_{A2} $	$ \mathbf{t} \cdot \mathbf{e}_{A2} $
$\mathbf{e}_{B1}$	$a_{A1} \mathbf{e}_{A1} \cdot \mathbf{e}_{B1}  + a_{A2} \mathbf{e}_{A2} \cdot \mathbf{e}_{B1} $	$a_{B1}$	$ \mathbf{t} \cdot \mathbf{e}_{B1} $
$\mathbf{e}_{B2}$	$a_{A1} \mathbf{e}_{A1} \cdot \mathbf{e}_{B2}  + a_{A2} \mathbf{e}_{A2} \cdot \mathbf{e}_{B2} $	$a_{B2}$	$ \mathbf{t} \cdot \mathbf{e}_{B2} $

Eine Überlappung der Rechtecke A und B liegt nicht vor, sobald sich ihre projizierten Intervalle auf einer der zu überprüfenden Projektionsachsen  $\mathbf{l}_k$  nicht überdecken. Der Pseudocode für diese Überprüfung ist in Abbildung 2.11 enthalten.

```

bool SchnittOBBOBB2D(Box A, Box B)
{
     $t = c_B - c_A$ ;
    if ( $|t \cdot e_{A1}| > a_{A1} + a_{B1}|e_{B1} \cdot e_{A1}| + a_{B2}|e_{B2} \cdot e_{A1}|$ ) then return false;
    if ( $|t \cdot e_{A2}| > a_{A2} + a_{B1}|e_{B1} \cdot e_{A2}| + a_{B2}|e_{B2} \cdot e_{A2}|$ ) then return false;
    if ( $|t \cdot e_{B1}| > a_{A1}|e_{A1} \cdot e_{B1}| + a_{A2}|e_{A2} \cdot e_{B1}| + a_{B1}$ ) then return false;
    if ( $|t \cdot e_{B2}| > a_{A1}|e_{A1} \cdot e_{B2}| + a_{A2}|e_{A2} \cdot e_{B2}| + a_{B2}$ ) then return false;
    return true;
}

```

Abbildung 2.11: Pseudocode Überlappungstest für beliebig orientierte Rechtecke

### 2.3.5.2 Überschneidungstest zwischen einer Strecke und einer orientierten Box

Die Schnittüberprüfung einer Strecke mit einer beliebig orientierten Box kann in Analogie zum Überlappungstest zweier orientierter Boxen auf Basis des *Separierenden Achsentheorems* erfolgen. Ein alternativer Ansatz wird in [135, 149] vorgestellt. Dabei finden Schnittberechnungen zwischen einem Strahl und den gegenüberliegenden, parallel verlaufenden Ebenen, die durch die Kanten des Quaders aufgespannt werden, für jede der drei Achsen statt. Anhand der Auswertung der Abstände zu den einzelnen Schnittpunkten kann festgestellt werden, ob es tatsächlich einen Schnitt mit den Seitenflächen des Quaders gibt. Bei der Verwendung des Separationsansatzes für eine Strecke und eine beliebig im Raum orientierte Box ergibt sich die Anzahl der zu testenden Achsen aus den 3 Flächenrichtungen der Box und den 3 Kombinationen der Kantenrichtungen der Box mit dem Richtungsvektor der Strecke zu 6. Eine Beschreibung dieser Tests befindet sich in [42]. Durch die Verwendung einer Begrenzungsbox mit einer achsenparallelen Ausrichtung in der z-Koordinate vereinfacht sich der in dieser Arbeit vorgestellte Schnitttest. Dieser Test setzt sich aus zwei Teilen zusammen. Im ersten Teil des Testes findet eine Schnittüberprüfung der Strecke mit der oberen und unteren Ebene der Box statt. Das Ziel dieses Testes ist die Bestimmung des Streckenabschnittes, der sich zwischen diesen beiden Ebenen befindet. Mit diesem auf die x-y-Ebene projizierten Streckenabschnitt findet im zweiten Teil eine Schnittüberprüfung mit der Grundfläche der Box statt.

Bei der Schnittüberprüfung der Strecke mit dem Rechteck werden drei Separationstests durchgeführt, welche sich aus den zwei Kantenrichtungen des Rechtecks und der Richtung des Normalenvektors der Strecke ergeben.

Die Strecke ist durch eine Ursprungsposition  $\mathbf{a}$  und den Richtungsvektor  $\mathbf{v}$ , der im Betrag der Streckenlänge entspricht, definiert. Als Hilfsgröße wird der Mittelpunkt  $\mathbf{c}_s$  der Strecke und ein Richtungsvektor mit der halben Streckenlänge  $\mathbf{v}_{0.5}$  und dessen Normale  $\mathbf{v}_{0.5norm}$  eingeführt:

$$\begin{aligned}
 \mathbf{v}_{0.5} &= 0.5 \cdot \mathbf{v} \\
 \mathbf{v}_{0.5norm} &= [\mathbf{y}_{\mathbf{v}_{0.5}}, -\mathbf{x}_{\mathbf{v}_{0.5}}] \\
 \mathbf{c}_s &= \mathbf{a} + \mathbf{v}_{0.5}
 \end{aligned} \tag{2.26}$$

Für die ersten beiden Tests werden als Projektionsachsen die Kantenrichtungen des Rechtecks verwendet. Beim letzten Test ist die Projektionsachse die Normale  $\mathbf{v}_{0.5norm}$  des Richtungsvektors  $\mathbf{v}_{0.5}$ .

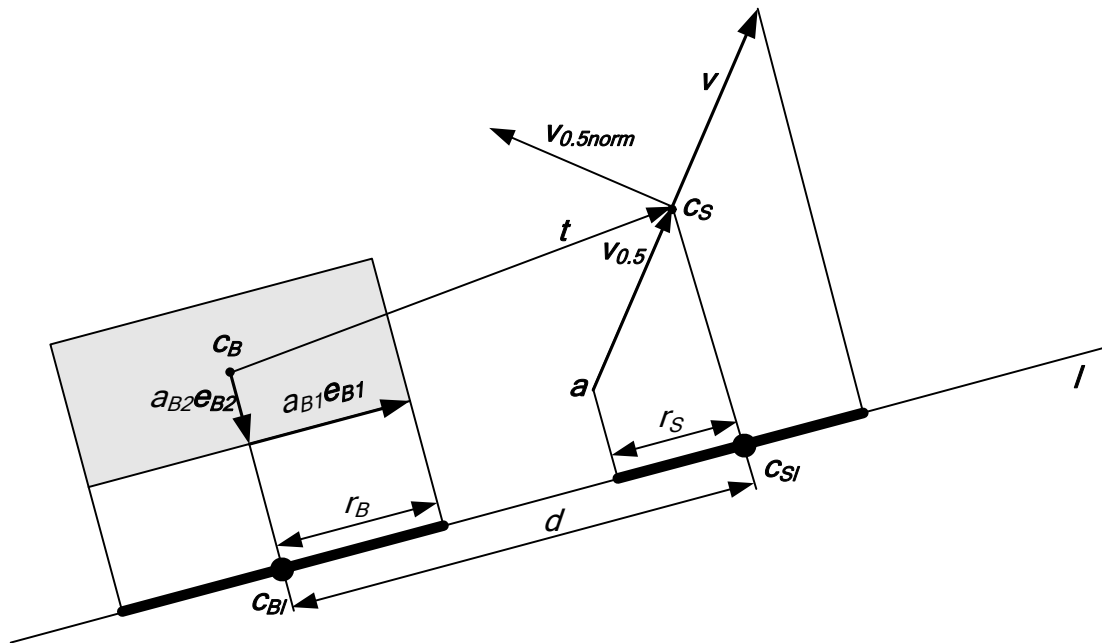


Abbildung 2.12: Separationstest einer Strecke und eines Rechteckes

Da dieser Vektor kein Einheitsvektor ist, entsprechen die Größen  $r_B$  und  $d$  nicht den realen Längen. Dies ist auch nicht erforderlich, da ihre Größenverhältnisse stimmen. (Die realen Längen erhält man, wenn die Größen durch den Betrag des Vektors  $v_{0.5norm}$  geteilt werden.) Die zu berechnenden Größen für die einzelnen Separationstests sind in Tabelle 2.2 aufgeführt. Der Pseudocode für diesen Schnittest ist in Abbildung 2.13 enthalten.

Tabelle 2.2: Bestimmung der Radien  $r_B$  und  $r_S$  und des projizierten Abstandes  $d$ 

Projektionsachse $l_k$	$r_B$ der Box	$r_S$ der Strecke	$d$
$e_{B1}$	$a_{B1}$	$ v_{0.5} e_{B1} $	$ t e_{B1} $
$e_{B2}$	$a_{B2}$	$ v_{0.5} e_{B2} $	$ t e_{B2} $
$v_{0.5norm}$	$a_{B1} e_{B1} v_{0.5norm}  + a_{B2} e_{B2} v_{0.5norm} $	0	$ t v_{0.5norm} $

```

bool SchnittSegmentOBB2D(Strecke S, Box B)
{
     $v_{0.5} = 0.5v$ ;
     $v_{0.5norm} = [y_{v0.5}, -x_{v0.5}]$ ;
     $c_S = a + v_{0.5}$ ;
     $t = c_S - c_B$ ;
    if ( $|t e_{B1}| > a_{B1} + |v_{0.5} e_{B1}|$ ) then return false;
    if ( $|t e_{B2}| > a_{B2} + |v_{0.5} e_{B2}|$ ) then return false;
    if ( $|t v_{0.5norm}| > a_{B1}|e_{B1} v_{0.5norm}| + a_{B2}|e_{B2} v_{0.5norm}|$ ) then return false;
    return true;
}

```

Abbildung 2.13: Pseudocode für den Schnittest zwischen einer Strecke und einem beliebig orientierten Rechteck

## 2.4 Laufzeitmessungen

In den nachfolgenden Abschnitten werden die Laufzeiten der einzelnen Algorithmen zur Kollisionserkennung untersucht. Dabei sollen, ausgehend von den einzelnen Schnitttests bzw. -berechnungen zwischen zwei Körpern, verschiedene mehrstufige Konzepte zur Kollisionserkennung auf ihre Performance überprüft werden.

Die durchgeführten Tests zur Ermittlung der Rechenzeiten werden auf einem Rechner mit einem Pentium III 600 MHz Prozessor durchgeführt. Dieser Prozessor entspricht hinsichtlich seiner Rechenleistung der im Fahrzeug eingesetzten Recheneinheit. Da die Berechnungszeiten der einzelnen Algorithmen sehr gering sind, ist die Verwendung eines Timers mit einer hohen Auflösung eine wichtige Voraussetzung für eine korrekte Messung. Aus diesem Grund wird der *High-Resolution Timer* eingesetzt [109]. Dieser Timer arbeitet mit einem hochauflösenden Performance-Zähler (*high-resolution performance counter*). Die Taktfrequenz zum Hochzählen des Zählers hängt von seiner hardwaremäßigen Implementierung ab. Bei einigen Systemen wird die Frequenz eines Timer Chips oder die Zykluszeit der Prozessoruhr verwendet. Für den in dieser Arbeit eingesetzten Rechner beträgt die Taktfrequenz des Zählers 3,579545 MHz, was einem Grundtakt von 0,279365  $\mu$ s entspricht. Zur Ermittlung der Taktfrequenz wird die Funktion *QueryPerformanceFrequency* verwendet. Die Anzahl der Takte seit dem Computerstart ermittelt die Funktion *QueryPerformanceCounter*. Sie wird zur Ermittlung der Start- und Stopzeit verwendet. Da der Grundtakt des Timers in der Größenordnung der Rechenzeit einiger zu untersuchender Algorithmen liegt, wird die Genauigkeit der Messung durch das  $n$ -fache Aufrufen des zu untersuchenden Algorithmus innerhalb einer Schleife erhöht. Die mittlere Rechenzeit wird dann aus der Teilung der Abarbeitungszeit der Schleife durch den  $n$ -mal aufgerufenen Algorithmus gebildet. Durch diese Maßnahme wird auch die zeitliche Verfälschung, die durch die Verarbeitungszeit bei der Abfrage des Zählers beim Start und bei der Beendigung entsteht, im Verhältnis zur Rechenzeit des  $n$ -mal aufgerufenen Algorithmus gesenkt. Des Weiteren setzen alle Testprogramme zu Beginn ihrer Abarbeitung ihren Thread auf die höchste Priorität. So können andere Prozesse das Testprogramm nicht unterbrechen, womit eine Verfälschung der Messung vermieden wird.

### 2.4.1 Laufzeiten zur Erzeugung der einzelnen Attribute

Bevor eine Durchführung der Hüllkörper-Schnitttests der Manöver und der Objekte erfolgen kann, müssen aus ihren geometrischen Daten die einzelnen Hüllkörper erzeugt werden. Diese Berechnungen brauchen für jedes Objekt und Manöver nur einmal durchgeführt zu werden. Für die Objekte geschieht dies bei Erhalt einer neuen Objektliste vom Modul *Objekterkennung*. Die Erzeugung der Manöver-Hüllkörper erfolgt nach Empfang der Manöverliste vom Modul *Manöververarbeitung*. Während der zyklisch aufgerufenen *Kollisionsüberwachung* müssen nur die Hüllkörper für das aktuelle Manöver an seinem Abarbeitungsgrad angepasst werden. Zusätzlich zur Erzeugung der Hüllkörper werden in dieser vorbereitenden Phase auch die Parameter zur Beschreibung der Objekte und Manöver gebildet. Diese Vorverarbeitung senkt den Rechenaufwand bei den eigentlichen Schnittberechnungen. Die mittleren Rechenzeiten zur Erzeugung der Attribute (Hüllkörper, Parameter der geometrischen Formen) für das Objekt und die Manöver sind in Abbildung 2.14 dargestellt. In diesem Diagramm ist zum Vergleich auch der prozentuale Anteil der Berechnungen bezogen auf die Updatezeit von 100 ms abgebildet.

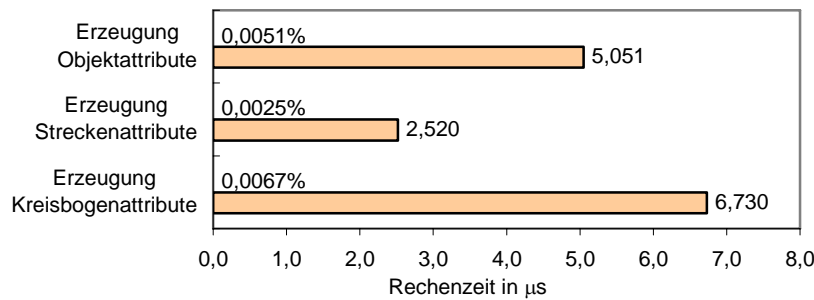


Abbildung 2.14: Mittlere Rechenzeiten zur Erzeugung der Attribute für ein Objekt und für die Manöver

## 2.4.2 Laufzeiten der einzelnen Überlappungs- bzw. Schnitttests

In Abbildung 2.15 sind die minimalen und maximalen Berechnungszeiten der einzelnen Schnittalgorithmen angegeben. Eine Unterscheidung zwischen minimaler und maximaler Rechenzeit resultiert aus dem Aufbau der Algorithmen. Die Algorithmen bestehen aus einzelnen Tests oder enthalten Fallabfragen, die zu einer vorzeitigen Beendigung des Algorithmus führen können. Die geringste Rechenzeit benötigt die Überprüfung der achsenparallelen Begrenzungsboxen (AABB/AABB) auf Überlappung. Hier sind ausschließlich Größenvergleiche durchzuführen. Die Rechenzeit  $t_{NP\_Strecke}$  für die korrekte Schnittberechnung zwischen der Strecke und dem elliptischen Zylinder (Strecke/Ellipt. Zylinder) entspricht gerade 72 % mehr als zwischen der Strecke und einem gerichteten Quader (Strecke/OBB)  $t_{WP2\_Strecke}$ . Danach muss nach Gleichung (2.22) die Schnittwahrscheinlichkeit  $P(SWP2\_Strecke)$  kleiner als 0,42 sein, damit der Einsatz des Strecke/OBB Tests effizient ist. Der große Rechenaufwand der Schnittberechnung zwischen Kreisbogen und elliptischem Zylinder erklärt sich durch die aufwendigen Rechenoperationen (siehe Abschnitt 2.3.2.2). Die Schnittwahrscheinlichkeit  $P(SWP2\_Arc)$  muss hier kleiner als 0,837 sein, damit die Verwendung des OBB/OBB Tests in der *Weiten Phase* des Kollisionskonzeptes zu einer Beschleunigung in der Abarbeitung führt.

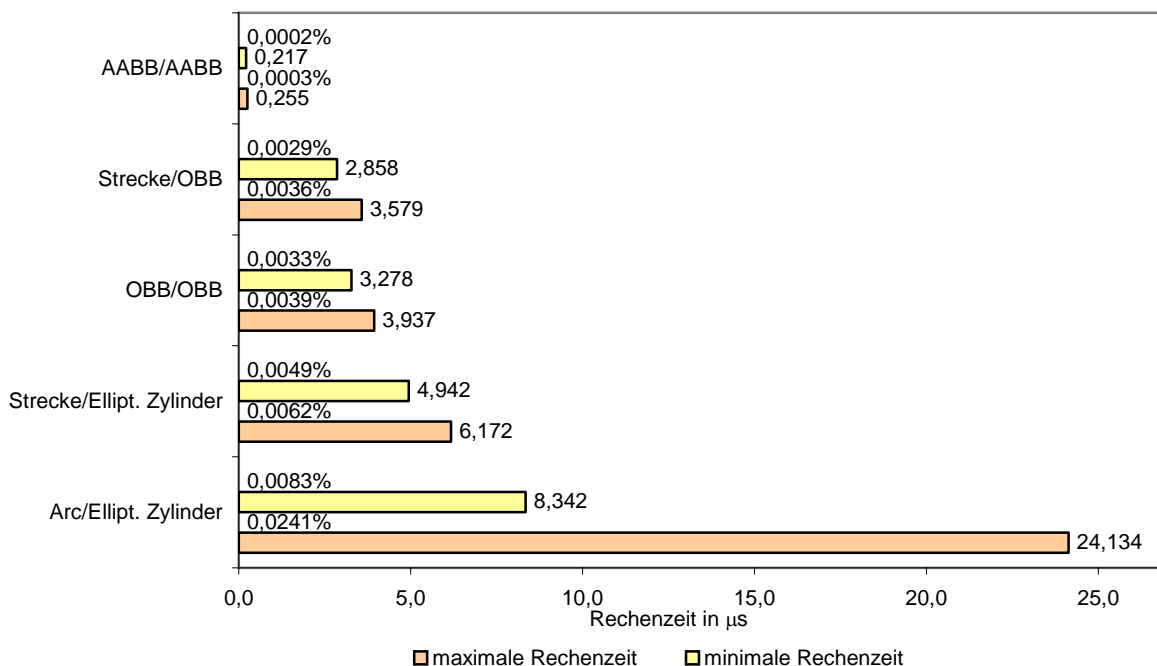


Abbildung 2.15: Rechenzeiten der einzelnen Schnittberechnungen zwischen Objekt und Manöver

### 2.4.3 Bestimmung der Schnittwahrscheinlichkeiten

Bei der Laufzeitauswertung der einzelnen Überlappungs- und Schnitttests des vorhergehenden Abschnittes wurden die Schnittwahrscheinlichkeiten  $P(S_{WP2\_Strecke})$  und  $P(S_{WP2\_Arc})$  für eine effiziente Arbeitsweise der zweiten Stufe der *Weiten Phase* bestimmt. Um diese Schnittwahrscheinlichkeiten für die Manöver Strecke und Kreisbogen mit einem Objekt zu ermitteln, werden Monte-Carlo-Simulationen durchgeführt.

Als repräsentative Objektmaße werden  $a_{Ellipse} = 15$  m,  $b_{Ellipse} = 5$  m und  $h = 20$  m gewählt. Bei den einzelnen Versuchen erfolgt eine zufällige Positionierung und Ausrichtung des elliptischen Zylinders und der Strecke bzw. des Kreisbogens im Operationsgebiet. Anschließend findet der Überlappungstest zwischen ihren achsengerichteten Begrenzungsboxen statt. Gab es eine Überlappung, wird der entsprechende Test der zweiten Stufe der *Weiten Phase* durchgeführt. Die Bestimmung von  $P(S_{WP2\_Strecke})$  erfolgt für unterschiedliche Streckenlängen  $l$  im Bereich von 15 m bis 210 m durch entsprechende Simulationen. Für den Kreisbogen wurde der Radius  $r$  im Bereich von 15 m bis 210 m variiert. Die Anzahl der durchzuführenden Versuche in einer Simulation bestimmt sich nach [52] zu:

$$n = \frac{1}{4\delta\varepsilon^2}. \quad (2.27)$$

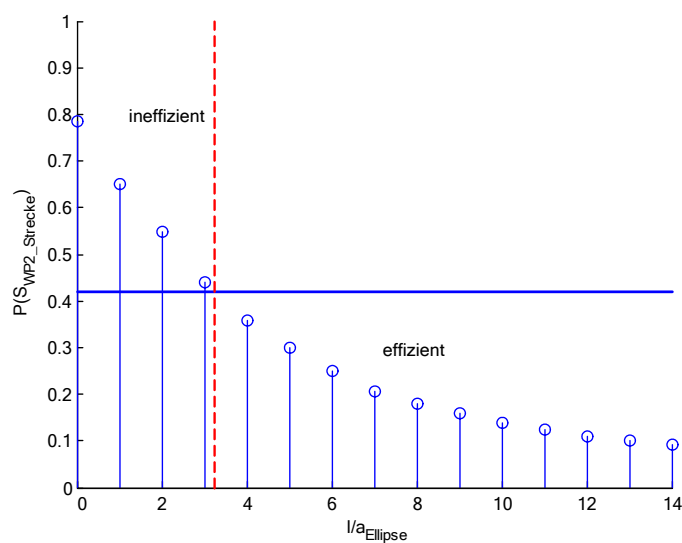
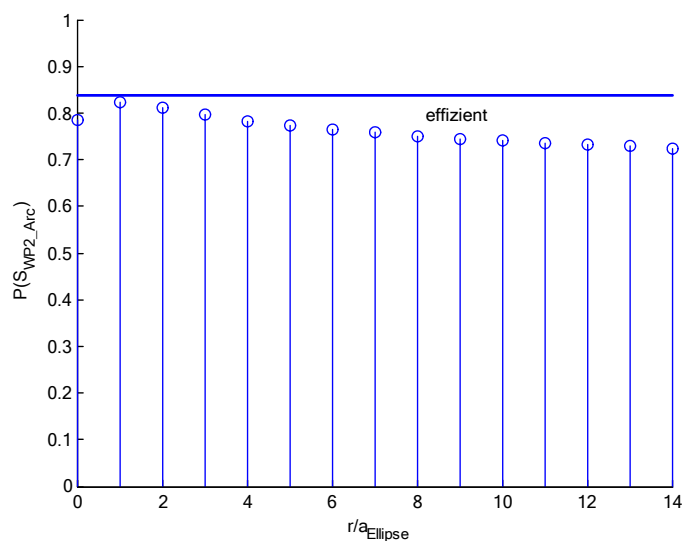
Die Irrtumswahrscheinlichkeit  $\delta$  wurde zu 0,01 gewählt. Das heißt, mit einer 99%igen Wahrscheinlichkeit sind  $1e+6$  Versuche notwendig, damit der vorgegebene Fehler  $\varepsilon$  zwischen dem empirisch ermittelten Wert  $\overline{P(S)}$  und dem exakten Wert  $P(S)$  kleiner als 0,005 ist. Die Schnittwahrscheinlichkeit  $\overline{P(S)}$  ermittelt sich durch Quotientenbildung aus der Anzahl der erfolgreichen Tests der zweiten Stufe  $n_{WP2\_pos}$  durch die Anzahl der erfolgreichen Tests der ersten Stufe  $n_{WP1\_pos}$  der *Weiten Phase*. Unter erfolgreich versteht man einen vorhandenen Schnitt bzw. eine vorhandene Überlappung im durchgeführten Versuch. Damit gilt:

$$\overline{P(S)} = \frac{n_{WP2\_pos}}{n_{WP1\_pos}} = \frac{n_{NP}}{n_{WP2}}. \quad (2.28)$$

In Abbildung 2.16 und Abbildung 2.17 sind die ermittelten Schnittwahrscheinlichkeiten in Abhängigkeit der Verhältnisse  $l/a_{Ellipse}$  bzw.  $r/a_{Ellipse}$  dargestellt. Dabei zeigt die waagerechte Linie die berechnete Schnittwahrscheinlichkeit für eine effiziente Arbeit der zweiten Stufe. Da für den Kreisbogen die simulierten Schnittwahrscheinlichkeiten stets unterhalb der Linie liegen, ist durch den Einsatz der zweiten Stufe immer eine Performancesteigerung gegeben. Für ein Verhältnis  $l/a_{Ellipse}$  kleiner als 3,25 führt der Einsatz der zweiten Stufe zu keiner Performancesteigerung. Hier kann die zweite Stufe entfallen. Die Schnittwahrscheinlichkeit für  $l/a_{Ellipse}$  bzw.  $r/a_{Ellipse} = 0$  wurde durch Geometrische Wahrscheinlichkeit nach der Formel:

$$P(S) = \frac{\pi \cdot a_{Ellipse} b_{Ellipse}}{2 \cdot a_{Ellipse} 2 \cdot b_{Ellipse}} = \frac{\pi}{4} \quad (2.29)$$

bestimmt. Dabei beschreibt der Dividend die Fläche der Ellipse und der Divisor die Fläche der orientierten Begrenzungsbox dieser Ellipse.

Abbildung 2.16: Schnittwahrscheinlichkeit  $P(S_{WP2\_Strecke})$  in Abhängigkeit von  $l/a_{\text{Ellipse}}$ Abbildung 2.17: Schnittwahrscheinlichkeit  $P(S_{WP2\_Arc})$  in Abhängigkeit von  $r/a_{\text{Ellipse}}$ 

## 2.4.4 Vergleich unterschiedlicher Konzepte

Nachfolgend sollen mehrere Konzepte zur Kollisionserkennung auf ihre Performance mit den in Tabelle 2.3 beschriebenen Missionsszenarien untersucht werden.

Tabelle 2.3: Missionsszenarien

Test	Anzahl der Manöver			Anzahl der Objekte
	Arc	Strecke	$\Sigma$	
Test 1	11	12	23	24
Test 2	11	12	23	48
Test 3	24	25	49	48
Test 4	37	38	75	72
Test 5	50	51	101	96
Test 6	63	64	127	120

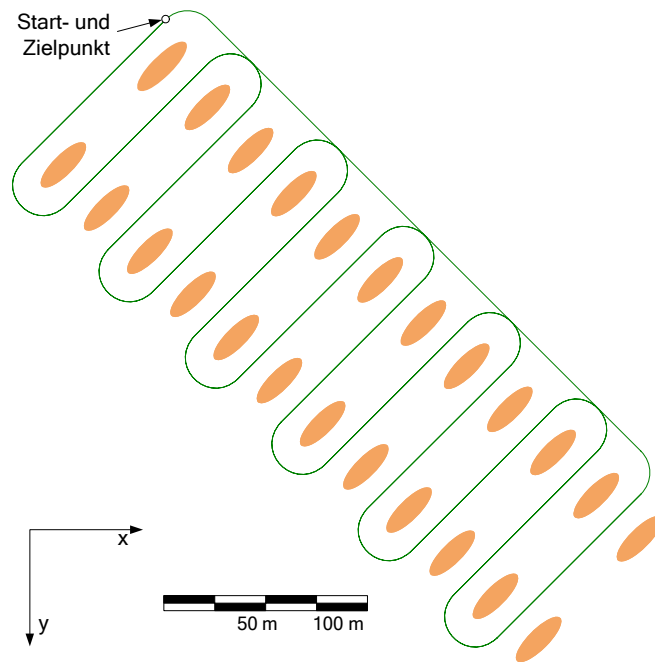


Abbildung 2.18: Missionsszenario

Die Manöver verlaufen mäanderförmig um die einzelnen Objekte herum (siehe Abbildung 2.18). Für die Tests 3 bis 6 wurden die Objekte und die Manöver in z-Koordinatenrichtung übereinander angeordnet. Test 2 entspricht hinsichtlich der Manöver- und Objektanzahl den Anforderungen an das Submodul *Kollisionsüberwachung*.

Abbildung 2.19 zeigt die Rechenzeiten zur Erzeugung der einzelnen Attribute. Die Gesamtberechnungszeit für Test 2 beträgt 0,4 ms. Da die Objektliste und die Manöverliste nicht immer gleichzeitig übergeben werden, teilt sich die Gesamtberechnungszeit auf zwei Updateroutinen auf.

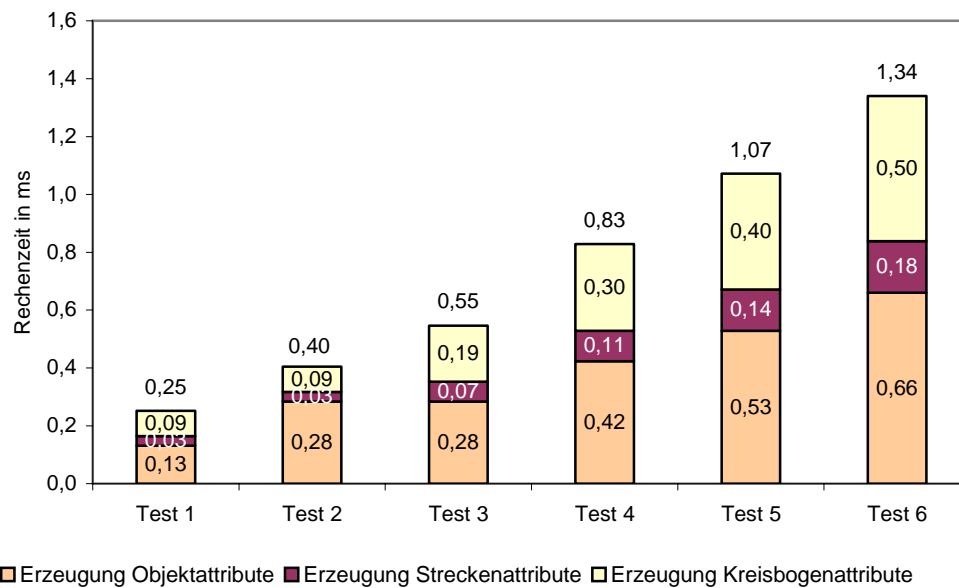


Abbildung 2.19: Berechnungszeit der Attribute für die einzelnen Manöver und Objekte bei den einzelnen Tests



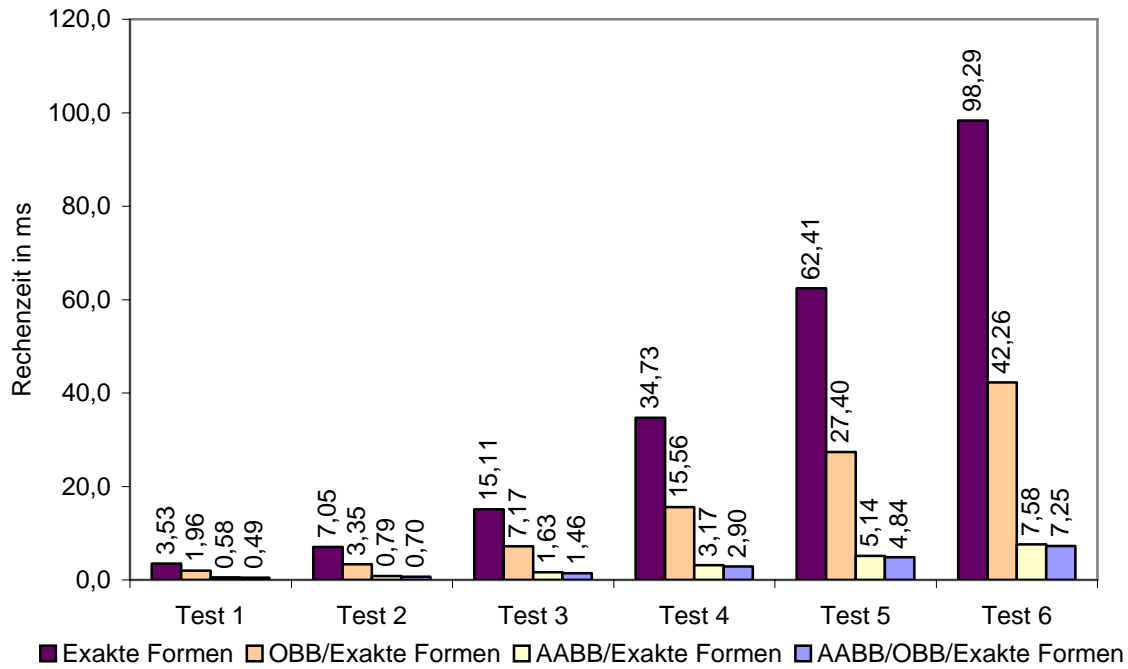


Abbildung 2.20: Ergebnisse der Laufzeittests für die unterschiedlichen Konzepte zur Kollisionserkennung

In Abbildung 2.20 sind die Rechenzeiten für die einzelnen Tests der untersuchten Konzepte angegeben. Dabei zeigt sich der Vorteil des Zwei-Phasen-Konzeptes gegenüber einer direkten Durchführung der Schnittberechnungen (Exakte Formen). Der Einsatz der ersten Stufe der *Weiten Phase* bringt die größte Beschleunigung (AABB). Hier können durch einfache und schnelle Tests viele Kollisionspaare aussortiert werden, welche nicht kollidieren. Wird in der *Weiten Phase* nur mit den orientierten Begrenzungsboxen (OBB/Exakte Formen) gearbeitet, ist der Rechenaufwand 4- bis 5-mal größer als bei einer Verwendung von achsengerichteten Begrenzungsboxen (AABB/Exakte Formen). Die zusätzliche Verwendung der zweiten Stufe (OBB) der *Weiten Phase* bringt für das Missionsszenario Test 2 noch einmal eine Beschleunigung von 12 %. So beträgt die Rechenzeit 0,7 ms. Diese Zeit wird dann erreicht, wenn die gesamte Manöverliste überprüft werden muss und es keine frühzeitige Beendigung durch eine Kollision gibt. Die prozentuale Zeitersparnis  $T_{\%}$  der zweiten Stufe kann für die einzelnen Manöver unter Verwendung von Gleichung (2.19) und (2.20) zu

$$T_{\%} = 100 \% \cdot \left( \left( 1 - P(S_{WP2}) \right) - \frac{t_{WP2}}{t_{NP}} \right) \quad (2.30)$$

ermittelt werden. Für die geometrischen Abmaße der Objekte und Manöver des hier definierten Missionsszenarios würde sich eine theoretische Zeitersparnis von  $T_{\%\_Strecke} = 19 \%$  und  $T_{\%\_Arc} = 1,4 \%$  ergeben. Das entwickelte Zwei-Phasen-Konzept mit den zwei Stufen in der *Weiten Phase* führte bei allen Tests zu den geringsten Rechenzeiten. Ist jedoch mit anderen Größenverhältnissen zwischen den Manövern und den Objekten zu rechnen, ist der Einsatz der zweiten Stufe zu überprüfen. Die Verwendung der *Weiten Phase* führt neben der Beschleunigung in der Abarbeitung auch zu einer Erhöhung der numerischen Stabilität. Dies begründet sich daraus, dass weit entfernte Objekte ausgeschlossen werden, was nach Gleichung (2.7) und (2.15) zu einer Verkleinerung der Polynomkoeffizienten führt.

## 2.5 Zusammenfassung

In diesem Kapitel wurde das Konzept zur Kollisionsüberwachung vorgestellt. Die Bestimmung der Kollisionssituation erfolgte in einem Konfigurationsraum. Dabei wird das Fahrzeug als Punkt modelliert und eine Vergrößerung der einzelnen Objektformen durchgeführt. Die Kollisionssituation wird dann durch Schnittberechnungen der Manövertrajektorien mit den Elliptischen Zylindern der Objekte ermittelt. Jedes einzelne Manöver wird nach der Reihenfolge seiner Abarbeitung im Missionsplan gegen die gesamte Liste der Objekte geprüft, was einem simulierten Abfahren der Mission entspricht.

Die Manöver und die Objekte lassen sich durch Kurven maximal 2. Ordnung beschreiben, so dass analytische Schnittberechnungen durchgeführt werden können. Zur Einhaltung der Echtzeitforderung bei der vorgegebenen maximalen Anzahl an zu überprüfenden Manövern und Objekten wurde ein Zwei-Phasen-Algorithmus verwendet. In der *Weiten Phase* werden die Manöver und Objekte durch Hüllkörper nachgebildet, welche durch einfache Tests auf Überlappung überprüft werden. In der *Nahen Phase* finden dann die eigentlichen Schnittberechnungen mit den in der *Weiten Phase* ermittelten Kollisionskandidaten statt. Im Gegensatz zu der in vielen Ansätzen verwendeten hierarchischen Unterteilung der Hüllkörper in eine Baumstruktur, findet in dieser Arbeit lediglich eine zweistufige Unterteilung in der *Weiten Phase* statt. Dabei werden die zu überprüfenden geometrischen Formen in den einzelnen Stufen durch unterschiedliche Hüllkörper nachgebildet. Der Wechsel der Hüllkörperform führt zur schnellen Vorselektion der potenziellen Kollisionskandidaten. Dies ist vor allem beim Streckenmanöver ersichtlich, wo in der zweiten Stufe der *Weiten Phase* schon die reale Manöverform für den Schnitttest verwendet wird. In der ersten Stufe der *Weiten Phase* werden als Hüllkörper achsengerichtete Boxen (AABB) zur Nachbildung der Manöver und Objekte eingesetzt. Die Effizienz der zweiten Stufe in der *Weiten Phase* wurde unter Verwendung von Schnittwahrscheinlichkeiten auf Basis der Rechenzeiten und der Größenverhältnisse zwischen Manöver und Objekt untersucht. Es konnte gezeigt werden, dass ein mehrstufiges Konzept nicht immer zu einer Beschleunigung in der Abarbeitung führt.

Die Kollisionsüberprüfung erfolgt für jedes Manöver mit allen detektierten Objekten. Diese Vorgehensweise ist einfach und erfüllt bei der möglichen Manöver- und Objektanzahl die Laufzeitanforderungen (siehe Abschnitt 2.4). Müssen größere Objektmengen überprüft werden, kann der Einsatz eines *Sweep and Prune Algorithmus* [112] zu einer Performancesteigerung führen. Dabei muss jedoch der Mehraufwand zur Erzeugung der dazu benötigten Datenstrukturen und seine Abarbeitung im Vergleich zur einfachen Überprüfung gesehen werden. Eine andere Möglichkeit wäre die Teilung des Operationsgebietes in einzelne Sektoren. Dadurch müssen nur die Manöver und Objekte innerhalb der einzelnen Sektoren auf Überschneidungen überprüft werden.

## 3 Kollisionsvermeidungssystem

Tangieren während einer Mission Objekte den abzufahrenden Missionsplan und besteht die aktuelle Missionsaufgabe im Ausweichen dieser Objekte, wird das Submodul *Kollisionsvermeidungssystem* aktiviert. Dieses System führt das Fahrzeug an den Objekten vorbei und zum Missionsplan zurück. Solche Objekte können Wracks, Minen, Trägerkonstruktionen oder natürliche Erhebungen sein, bei denen eine Kollision die Beschädigung bzw. die Zerstörung des Fahrzeuges zur Folge hätte. Der beim Ausweichen anzufahrende Rendezvouspunkt auf dem Missionsplan wird durch das Submodul *Zielpunktgenerierung* vorgegeben. Das Kollisionsvermeidungssystem berücksichtigt die Manövrierfähigkeit des Fahrzeuges und die vorhandenen Umweltbedingungen bei der Generierung seiner Führungsvorgaben für das Submodul *Führungsalgorithmen*. Die durchgeführten Ausweichmanöver sind dadurch gekennzeichnet, dass sie das Fahrzeug sicher um die Objekte navigieren und dabei zeit- bzw. energieoptimal sind.

Zu Beginn dieses Kapitels findet eine kurze Einführung in die Kollisionsvermeidung anhand einiger Begriffsbestimmungen und eines Vergleiches des reaktiven und des planenden Ansatzes statt. Im Anschluss werden bekannte Architekturen von Kollisionsvermeidungssystemen vorgestellt und bewertet. Nach einer Anforderungsbeschreibung des Kollisionsvermeidungssystems für das Modul *FIS* erfolgt eine detaillierte Vorstellung seiner neu entwickelten Architektur und Arbeitsweise.

### 3.1 Einführung in die Kollisionsvermeidung

Eine sichere Kollisionsvermeidung ist eine grundlegende Anforderung an ein Autonomes Mobiles System. Sie ist notwendig, wenn ein autonomes System in einer unbekannten oder teilweise bekannten Umwelt sicher agieren soll. In Abhängigkeit vom Einsatzgebiet, der verwendeten Sensorik, der zur Verfügung stehenden Rechentechnik und dem Bewegungsverhalten des Fahrzeuges sind in den letzten Jahrzehnten eine Vielzahl von Verfahren zur Kollisionsvermeidung entstanden. Mit dieser Vielzahl entstand auch eine unüberschaubare Begriffsvielfalt. Deshalb sollen einige in dieser Arbeit verwendete Begriffe im Abschnitt 3.1.1 kurz erläutert werden. Anschließend erfolgt ein Vergleich des reaktiven und des planenden Ansatzes zur Lösung der Kollisionsvermeidungsaufgabe.

#### 3.1.1 Begriffsbestimmung

Auf dem Gebiet der Autonomen Mobilen Systeme hat sich eine Begriffsvielfalt herausgebildet, welche dem Interessenten keinen leichten Zugang ermöglicht, um die vorgestellten Verfahren und Methoden richtig einordnen und miteinander vergleichen zu können. So werden für ein und dieselbe Methode verschiedene Begriffe verwendet. Andere Begriffe werden in unterschiedliche Zusammenhänge gebracht, so dass es zu Mehrdeutigkeiten kommt.

Tabelle 3.1: Verwendete Begriffe aus dem Gebiet der Autonomen Mobilen Systeme

		In dieser Arbeit gebrauchter Begriff	Synonyme
Aufgabe		Kollisionsvermeidung [31], [114] (engl. collision avoidance) [54]	Navigationsaufgabe [19], [73], [80] Aktionssteuerung [31], [45] Punkt-zu-Punkt-Bewegung [71], [77], [101] Bewegungssteuerung [5] Bewegungsplanung [4], [25], [61], [88], [98] Hindernisvermeidung [2], [4], [114] (engl. obstacle avoidance) [35], [90], [102], [122]
Lösungskonzepte	Planender Ansatz	Wegeplanung [25], [73], [88]	Pfadplanung [19], [113] Geometrische Bahnplanung [157] Globale Bahnplanung [101], [113] (engl. global path planning) [152] Bahnplanung [77], [98], [126], [133] (engl. path planning) [78], [122], [123] Wegplanung [31], [80], [92] Globale Routenplanung [60] Routenplanung [5], [68] Vorausplanung [98] (engl. pre-planning) Modellbasierte Ansätze [5], [126] Routenbasierte Ansätze [5] Weltmodellierung [4]
	Reaktiver Ansatz	Reaktive Steuerung [147]	Bewegungssteuerung [19] Lokale Bewegungssteuerung [157] Lokale Kollisionsvermeidung [71], [101] Reflexive Kollisionsvermeidung [31] Reaktive Planung [98] (engl. reactive planning) Verhaltensbasierte Ansätze [2], [5], [147] (engl. behavior-based) [6], [121] Verhaltensbasierte Steuerung [4]

Tabelle 3.1 gibt einen Überblick über die Begriffsvielfalt auf dem Gebiet der Autonomen Mobilen Systeme anhand der in dieser Arbeit verwendeten Begriffe. Vielfach enthalten die Namen der Lösungskonzepte in dieser Tabelle die Attribute *lokal* und *global*. Damit soll das Gebiet der verwendeten Hindernisinformationen und die Reichweite der generierten Handlungen besser charakterisiert werden. Bei den nachfolgenden Definitionen erfolgt eine Trennung zwischen der Aufgabe (Definition 1) und ihren möglichen Lösungskonzepten (Definition 2 und Definition 3). Die Aufgabe des in diesem Kapitel vorgestellten Systems ist die Kollisionsvermeidung, die in dieser Arbeit durch die folgende Definition beschrieben wird:

**Definition 1 (Kollisionsvermeidung):** *Die Aufgabe der Kollisionsvermeidung besteht in der Führung eines mobilen Systems von einem Startpunkt, welcher häufig die gegenwärtige Position des Systems darstellt, zu einem vorgegebenen Zielpunkt, ohne dabei mit Hindernissen im Operationsgebiet zu kollidieren.*

Der gewählte Begriff der Kollisionsvermeidung beschreibt nach Ansicht des Autors im Vergleich zu den in Tabelle 3.1 enthaltenen Synonymen am treffendsten die durchzuführende Aufgabe nach Definition 1. Zur Lösung der Kollisionsvermeidung gibt es zwei grundsätzlich verschiedene Konzepte. Diese Konzepte basieren auf einem planenden und einem reaktiven Ansatz. In dieser Arbeit werden dafür die Begriffe *Wegeplanung* und *Reaktive Steuerung* verwendet und durch die folgenden Definitionen beschrieben:

**Definition 2 (Wegeplanung):** Bei der Wegeplanung wird unter Verwendung aller Informationen des Operationsgebietes ein Weg von einem Startpunkt zu einem Zielpunkt erstellt, auf welchem das Fahrzeug geführt wird. Häufig werden bei der Planung des Weges neben der Grundforderung nach Kollisionsfreiheit auch Optimalitätskriterien, wie Zeit- bzw. Energieminimierung einbezogen. Die Wegeplanung setzt sich immer aus einer planenden und einer ausführenden Phase zusammen, wobei sie dadurch ein vorausschauendes Verhalten aufweist.

**Definition 3 (Reaktive Steuerung):** Die Reaktive Steuerung erzeugt direkt Führungsanweisungen für das Fahrzeug aus den Informationen des unmittelbaren Operationsgebietes und des anzufahrenden Zielpunktes. Dabei besteht eine enge Kopplung zwischen den wahrgenommenen Sensorinformationen und den generierten Steuerkommandos. Die reaktive Steuerung besitzt ein kurzsichtiges Verhalten ohne Einbeziehung der bisherigen ausgeführten Aktionen und einer Vorausschau.

### 3.1.2 Vergleich von planendem und reaktivem Ansatz

In diesem Abschnitt sollen die beiden Ansätze zur Kollisionsvermeidung anhand ihrer repräsentativen Eigenschaften vorgestellt und verglichen werden. Es sei noch einmal angemerkt, dass es eine Vielzahl von Verfahren in den beiden Gruppen gibt, die sich hinsichtlich ihrer Lösungsansätze und damit ihrer Leistungsfähigkeit unterscheiden. Des Weiteren gibt es Verfahren zur Kollisionsvermeidung, die sich nicht eindeutig zu einer der beiden Gruppen nach den im Abschnitt 3.1.1 vorgestellten Definitionen zuordnen lassen. Deshalb erfolgt nur ein Abriss der charakteristischen Merkmale der beiden Ansätze, wie sie bei der Großzahl der Verfahren zutreffen und sich in der Fachliteratur etabliert haben. Auf eine Beschreibung der Funktionsweise sowie eine Vorstellung einzelner Verfahren wird verzichtet. Zur *Wegeplanung* und zur *Reaktiven Steuerung* wird ausführlich in den jeweiligen Kapiteln der Arbeit eingegangen.

Jede planende Tätigkeit setzt einen gewissen Kenntnisstand voraus, damit das erreichte Ergebnis mit dem geplanten Ziel übereinstimmt. So muss auch beim planenden Ansatz zur Kollisionsvermeidung das Operationsgebiet ausreichend erkundet sein, um eine erfolgreiche Planung durchführen zu können. Dazu wird das gesammelte Hinderniswissen über das Operationsgebiet in einer mathematisch beschreibbaren Form abgebildet. Für diese Darstellungsform werden unterschiedliche Bezeichnungen verwendet, wie Weltmodell [4], Planungsraum [71], Umgebungsmodell [19] oder Umweltmodellierung [80]. Auf der Grundlage des vorhandenen Hinderniswissens erfolgt dann die Generierung eines Weges, wobei auch vorgegebene Optimalitätskriterien wie Zeit- oder Energieminimierung berücksichtigt werden können. Des Weiteren ist es möglich, das nichtholonome Bewegungsverhalten von Fahrzeugen bei der Wegeplanung mit einzubeziehen. Eine solche Generierung erfordert in Abhängigkeit vom verwendeten Planungsansatz, der gewählten Darstellungsform und der Größe des Operationsgebietes eine entsprechende Bearbeitungszeit, in der das Fahrzeug durch fehlende Führungsinformationen nicht gesteuert werden kann. Dieses negative Verhalten besitzen die reaktiven Ansätze nicht. Da sie nur das Hinderniswissen im unmittelbaren Umfeld des Fahrzeuges nutzen, um daraus direkt Führungskommandos zu generieren, sind diese Berechnungen nicht sehr rechenintensiv und können in Echtzeit durchgeführt werden. So können diese Verfahren bei plötzlich auftauchenden Hindernissen durch entsprechende Führungskommandos schnell reagieren. Zudem sind sie dadurch robust gegenüber den Variationen der detektierten Hindernisstrukturen. Ein solches Variieren kann durch die entfer-

nungsabhängige Detektionsgüte der Sensorik oder durch Sensorungenauigkeiten entstehen. Einige reaktive Verfahren arbeiten mit den Sensorrohdaten, ohne sie in eine symbolische Form zu überführen. Dies führt zu einer Senkung der Rechenzeit und stellt geringere Anforderungen an die verwendete Sensorik. Durch das reaktive Verhalten können abrupte Steuerkommandos entstehen, denen ein Fahrzeug mit einer nichtholonomen Bewegungskarakteristik und einer vorhandenen Verzögerung nicht sofort folgen kann. Weiterhin zeigen die abgefahrenen Trajektorien vielfach keinen annähernd wegoptimalen Verlauf.

Wird ein planender Ansatz in Operationsgebieten eingesetzt, bei denen es keine Vorkenntnisse über mögliche Hindernisse gibt, ist der Einsatz einer leistungsfähigen und robusten Sensorik mit einer großen Reichweite zur Erfassung der Umwelt eine wichtige Voraussetzung für seine erfolgreiche Arbeit. So kann ein entsprechend großes Gebiet erfasst und in einem Umweltmodell abgebildet werden, auf dessen Basis dann ein Weg zu einem vorgegebenen Zielpunkt ermittelt wird. Nur wenn dieses vorausschauende Handeln möglich ist, kann der planende Ansatz seine volle Stärke zeigen, welche neben dem sicheren Agieren bei Sackgassen, engen Passagen und bei komplexen Hindernisszenarien in der Generierung von optimalen Fahrtrajektorien liegt. Ist der Auffassungsbereich der Sensoren geringer oder sind die Hindernisszenarien sehr komplex (indoor-Umgebung), kann eine sukzessive Gewinnung des notwendigen Hinderniswissens während des Umherfahrens im Operationsgebiet erfolgen. Diese Vorgehensweise setzt jedoch ein genaues Navigationssystem zur Abspeicherung der Absolutpositionen der Hindernisse in das Umweltmodell und zur Positionsbestimmung des Fahrzeuges voraus. Sind bewegte Hindernisse im Operationsgebiet vorhanden, ist ihre Einbindung in das Umweltmodell nur sinnvoll, wenn ihr zukünftiger Bewegungsverlauf bekannt ist. Ist dies nicht möglich, können sie bei der Planung nicht berücksichtigt werden, wodurch der geplante Weg unbrauchbar sein kann. Diese Ausführungen sollen zeigen, dass für den Einsatz eines planenden Ansatzes bestimmte Bedingungen vorhanden sein müssen, damit er erfolgreich arbeiten kann. Einige wichtige Eigenschaften des planenden und des reaktiven Ansatzes sind in Tabelle 3.2 aus der Einschätzung des Autors und den Ergebnissen einer durchgeführten Literaturrecherche zusammengefasst.

Tabelle 3.2: Vergleich der Ansätze zur Kollisionsvermeidung

Anforderungen	Wegeplanung (planender Ansatz)	Reaktive Steuerung (reaktiver Ansatz)
Verarbeitungsgeschwindigkeit/Echtzeitfähigkeit	■ ■ ■ ■ ■	■ ■ ■ ■ ■
Robustheit gegenüber Positions- und Geometrieschwankungen der detektierten Hindernisse	■ ■ ■ ■ ■	■ ■ ■ ■ ■
Eignung bei dynamischen Hindernissen	■ ■ ■ ■ ■	■ ■ ■ ■ ■
Optimalität der abgefahrenen Wege	■ ■ ■ ■ ■	■ ■ ■ ■ ■
Handhabbarkeit (Komplexität) der Algorithmen	■ ■ ■ ■ ■	■ ■ ■ ■ ■
Verhalten bei plötzlich auftauchenden Hindernissen	■ ■ ■ ■ ■	■ ■ ■ ■ ■
Eignung zur Selbstbefreiung beim Festsitzen (deadlocks)	■ ■ ■ ■ ■	■ ■ ■ ■ ■
Berücksichtigung des Bewegungsverhalten des Fahrzeuges (nichtholonomes Verhalten, Verlauf der generierten Fahrtroute)	■ ■ ■ ■ ■	■ ■ ■ ■ ■

Legende:

■ ■ ■ ■ ■	ausgezeichnet	■ ■ ■ ■ ■	sehr gut
■ ■ ■ ■ ■	gut	■ ■ ■ ■ ■	erfüllt
■ ■ ■ ■ ■	bedingt erfüllt	■ ■ ■ ■ ■	nicht erfüllt

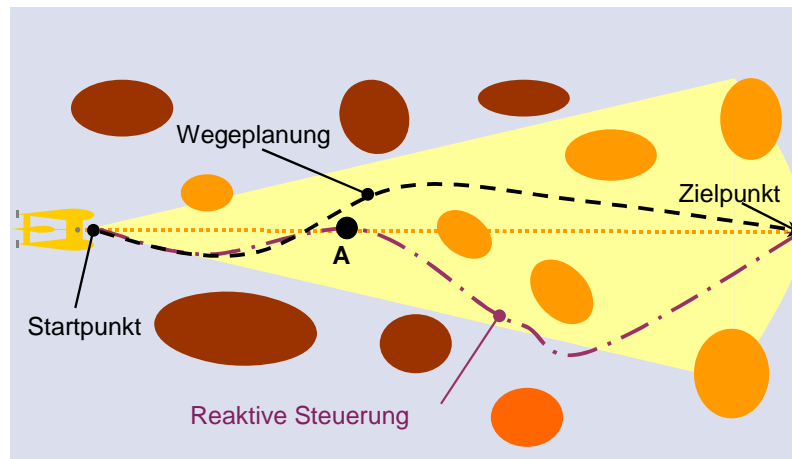


Abbildung 3.1: Fahrtrouten bei Verwendung der beiden Ansätze

Abbildung 3.1 stellt Fahrtrouten bei Verwendung des planenden und des reaktiven Ansatzes am Beispiel eines möglichen Hindernisparcours bei gegebenem Start- und Zielpunkt gegenüber. Es ist deutlich das reaktive Verhalten der *Reaktiven Steuerung* und das vorausschauende Verhalten der *Wegeplanung* an Punkt A zu erkennen.

### 3.1.3 Bekannte Realisierungen

Nachfolgend werden einige Verfahren zur Kollisionsvermeidung vorgestellt. Sie repräsentieren eine gezielte Auswahl, welche jedoch die unterschiedliche Herangehensweise bei der zu lösenden Kollisionsvermeidungsaufgabe in Abhängigkeit von der verwendeten Sensorik, den Einsatzbedingungen und den vorgegebenen Anforderungsschwerpunkten verdeutlicht. Bei der durchgeführten Literaturrecherche zeigte sich, dass vielfach durch eine Kombination beider Ansätze in einer gemeinsamen Architektur ihre speziellen Vorteile genutzt wurden.

In [71] wird ein hierarchisches Konzept zur Kollisionsvermeidung für einen mobilen Roboter vorgestellt. Dieses Konzept besteht aus einer Drei-Ebenen-Architektur. Die oberste Ebene, die Wegeplanung, bestimmt auf der Basis der gesamten vorhandenen Hindernisinformationen des Operationsgebietes und eines vorgegebenen Zielpunktes einen Routenvorschlag. Die Hindernisinformationen sind in einer globalen Datenbasis gespeichert, die ständig durch neu detektierte Hindernisse aktualisiert wird. Durch diese Veränderungen findet in der Wegeplanung fortwährend eine Planaktualisierung oder eine Neuplanung der Route zur Anpassung an die neue Umweltsituation statt. Für die mittlere Ebene, die Reaktive Steuerung, wird als Führungsgröße ein Zwischenzielpunkt vorgegeben, welcher sich aus der Schnittpunktberechnung der generierten Route mit dem Rand des Sensorauffassungsbereiches bestimmt. Die Reaktive Steuerung generiert unter Verwendung der aktuellen Sensorinformationen und des Zwischenzielpunktes eine Bewegungsreferenz für die untere Ebene, die Robuste Bewegungsregelung. Diese Bewegungsregelung sorgt dafür, dass das Fahrzeug der vorgegebenen Bewegungsreferenz genau folgt. Die Ausgangsgrößen der Regelung sind die Steuerkommandos für die Antriebe des Roboters. Diese Architektur benutzt einen planenden und einen reaktiven Ansatz zur Erfüllung der Kollisionsvermeidung. Die von der Wegeplanung generierte Route wird immer über die Reaktive Steuerung zur Ausführung gebracht. Schneidet die geplante Route ein Hindernis, so wird in der Reaktiven Steuerung ein neu anzufahrender Zwischenzielpunkt im nächsten hindernisfreien Sektor ermittelt. Da diese Route mit dem nicht mehr aktuellen Hinderniswissen generiert wurde, kann es zu einem konfusem Ausweichmanöver kommen.

Eine rückgekoppelte zweistufige Struktur für einen mobilen Roboter wird in [80] beschrieben. In der planenden Stufe, dem sogenannten *Navigator*, wird aus den vorhandenen Hindernisinformationen ein Wegenetz, bestehend aus Wegpunkten und Wegstrecken, erzeugt. Anschließend wird mit einem Suchalgorithmus der kürzeste Weg zum Zielpunkt ermittelt. Das Ergebnis ist eine Liste der abzufahrenden Wegstrecken. Mit diesen Wegstrecken findet abschließend eine Überprüfung auf Befahrbarkeit durch den Roboter statt. So dürfen diese Wegstrecken nicht zu nah an den Hindernissen liegen oder durch zu enge Passagen führen. Ggf. wird eine nicht befahrbare Wegstrecke durch zwei neue befahrbare Wegstrecken ersetzt. Wenn eine solche Modifikation nicht möglich ist, wird sie aus dem Wegenetz entfernt und eine neue Wegsuche initiiert. In einer zweiten Stufe, dem ausführenden Teil des *Navigators*, wird die aktuelle und die nächste Wegstrecke aus der Liste entnommen und nach einem Befahrbarkeitstest mit den aktuellen Hindernisinformationen an den *Piloten* übergeben. Ist eine Wegstrecke nicht befahrbar, findet eine Rückkopplung zur ersten Stufe des *Navigators* statt. Dort wird aus dem Wegenetz die nicht befahrbare Wegstrecke entfernt. Des Weiteren findet eine Modifikation des Wegenetzes unter Einbeziehung der neuen Hindernisinformationen statt. Abschließend wird der Suchalgorithmus gestartet. Liegt auf der aktuellen Wegstrecke ein Hindernis, werden die an den *Piloten* übergebenen Wegstrecken zurückgenommen. Der *Pilot* hält daraufhin das Fahrzeug an. Wenn erst auf der nächsten Wegstrecke ein Hindernis liegt, verlangsamt der *Pilot* die Fahrzeuggeschwindigkeit. Wird vor Erreichen der nächsten Wegstrecke ein neuer Weg gefunden, erhält der *Pilot* die neuen Wegabschnitte und beschleunigt wieder, ansonsten stoppt das Fahrzeug an der Endposition der aktuellen Wegstrecke. Ein solches Konzept ist auf Sicherheit bedacht, führt aber bei unvollständigem Hinderniswissen und fehlender Rechenleistung zu einer stockenden Bewegung durch das Operationsgebiet.

In [54] wird ein Kollisionsvermeidungssystem für ein AUV vorgestellt, welches aus drei Ebenen besteht. Die oberste Ebene enthält eine Wegeplanung zur Ermittlung eines energieoptimalen globalen Pfades. Dabei werden die Informationen von Seekarten und geloggtten Hindernisdaten zur Berechnung des Pfades verwendet. Dieser globale Pfad ist aus einzelnen Wegpunkten aufgebaut, welche die anzufahrenden Zielpunkte für die zweite Ebene bilden. Diese Ebene besteht aus einer lokalen Wegeplanung. Sie berechnet den kürzesten Weg zum nächsten Wegpunkt unter Verwendung der im Sonarbereich aufgefassten und in der Umgebung des AUVs befindlichen, schon bekannten Hindernisse. Dabei muss der Algorithmus kurze Bearbeitungszeiten sicherstellen, um auf neue Hindernisinformationen entsprechend schnell zu reagieren und einen neuen abzufahrenden Weg zur Verfügung zu stellen. Die dritte Ebene enthält eine Reaktive Steuerung, welche ein Ausweichmanöver initiiert, wenn sich Objekte zu nah am Fahrzeug befinden. Dies kann bei einer zu späten Objektdetektion infolge eines geringen Echsignals des Objektes, bei der Generierung eines Weges unter Verwendung falsch geloggtter Hindernispositionen oder während seines Abfahrens durch fehlerhafte Navigationsdaten passieren. Die oberste Ebene generiert einen neuen Weg, wenn die mittlere Ebene keinen Weg aus einer vorhandenen Hinderniskonstellation findet, die Abweichung zum vorgegebenen globalen Weg zu groß ist oder der Energieverbrauch niedrig gehalten werden soll. Dieses Konzept zur Kollisionsvermeidung zeigt die Probleme und Schwerpunkte im Unterwasserbereich, die durch das Abfahren großer Entfernungen mit begrenzten Energieressourcen und durch eingeschränkte Möglichkeiten zur Positionsbestimmung und zur Hinderniserkennung charakterisiert sind.



## 3.2 Anforderungen an die Kollisionsvermeidung

### 3.2.1 Objektdefinition

Das Modul *Objekterkennung* generiert ein symbolisches Abbild der vom Sonar detektierten Objekte des Operationsgebietes in Form von elliptischen Zylindern. Diese Zylinder werden in einer Liste an das *Kollisionsvermeidungssystem* übergeben. Durch das bei kleinen Objekten vorhandene schwache Echosignal und die Bearbeitungszeit der Sonarinformationen (siehe Abschnitt 1.2.2) kann es vorkommen, dass detektierte Objekte sehr spät an das *Kollisionsvermeidungssystem* gemeldet werden (ca. 30 m vor dem Fahrzeug). Des Weiteren können die übergebenen Objektdaten durch die entfernungsabhängige Detektionsgüte des Sonars hinsichtlich ihrer Abmaße und Position in bestimmten Grenzen variieren. Die Wahrscheinlichkeit einer Kollision mit bewegten Objekten wird in der Spezifikation zur Objekterkennung [79] als sehr gering eingeschätzt, aber bei der Erstellung des Anforderungskataloges für das Kollisionsvermeidungssystem auch nicht ausgeschlossen. Die zusätzliche Verwendung von geloggtten Hindernisdaten der bisherigen oder früheren Missionen ist in erster Linie von der Positioniergenauigkeit des verwendeten Navigationssystems abhängig. Eine Positionsbestimmung mittels GPS (engl.: Global Positioning System) ist unter Wasser durch die hohe Dämpfung des Wassers nicht möglich. Deshalb erfolgt die Positionsbestimmung durch ein Inertialnavigationssystem. Des Weiteren können akustische, magnetische oder visuelle Referenzmarken auf dem Meeresboden zur Positionsstützung eingesetzt werden [54]. Nach [54] wird die so erreichte Navigationsgenauigkeit mit 50 m pro Missionsstunde angegeben. Die erreichbare Genauigkeit bei Verwendung eines hochgenauen Inertialnavigationssystems beträgt nach [146] 39 m pro Stunde (0,5 sm nach 24 Stunden). Beim AUV DeepC wird ein Inertialnavigationssystem der Firma iMAR [63] eingesetzt, welches zusätzlich zu dem optischen Kreiselssystem die Sensordaten eines Doppler Velocity Logs (DVL) mit in die Positionsbestimmung einbezieht. Durch diese Datenfusion können Positioniergenauigkeiten von 0,2 % des zurückgelegten Weges erreicht werden [145]. Das entspricht einem Positionsfehler von 4 m nach einer Stunde Fahrt mit 2 m/s. Auch bei einer solchen Positioniergenauigkeit kann der Versatz zwischen der geloggtten und der realen Hindernisposition so groß werden, dass ein erzeugter Routenplan unbrauchbar ist.

### 3.2.2 Operationsgebiet

Bei einigen Einsatzfällen des AUVs ist die Verwendung von geloggtten Hindernisdaten recht unwahrscheinlich. In diesen Fällen geht es darum, ein unbekanntes, weiträumiges Gebiet von mehreren 100 km zu durchfahren und zu kartographieren. Die Wahrscheinlichkeit, einen Bereich während einer Mission wiederholt zu durchfahren, ist somit sehr gering. Für das systematische Durchsuchen eines Gebietes unter Verwendung eines Mäandermanövers können bei entsprechender Wahl des Sonarauffassbereiches auch seitlich des aktuellen Streckenabschnittes gelegene Objekte erfasst und geloggt werden. Diese Objekte können dann beim Abfahren der zukünftigen Streckenabschnitte mit in die Planung der Ausweichmanöver einbezogen werden. Da die Streckenabschnitte nebeneinander liegen, ist die zeitliche Verzögerung bzw. die zurückgelegte Strecke zwischen Erfassung und Nutzung der Hindernisinformationen gering und damit auch ihr Positionsfehler, welcher mit der zurückgelegten Wegstrecke korreliert. Aus diesen Überlegungen wurde ein Operationshorizont von 200 m Länge, 100 m

Breite und 100 m Höhe definiert, in dem die Ausweichmanöver durchgeführt werden. Die Anzahl an detektierten Objekten beträgt in diesem Gebiet maximal 50. Dies resultiert aus dem vorgegebenen Sonarauffassungsbereich, der möglichen Objektausdehnung und dem vorhandenen Umweltszenario sowie einem zusätzlichen Reservefaktor (50 % der real möglichen Objektanzahl). Somit kann zusammenfassend festgestellt werden, dass sich das Fahrzeug weitestgehend in unbekannter Umgebung bewegt, wobei vereinzelt mit Objekten zu rechnen ist. Ein sukzessiver Aufbau eines Umgebungsmodells, wie er bei indoor-Anwendungen oft praktiziert wird, kann durch die möglichen Missionsaufgaben in Verbindung mit der Größe des Operationsgebietes und des Navigationssystems nur begrenzt durchgeführt werden.

### 3.2.3 Manövrierfähigkeit und Umweltbedingungen

Das nichtholonome Bewegungsverhalten und die Dynamik der verwendeten AUVs führen zur Beschränkung ihrer Manövrierfähigkeit. An dieser Stelle sei noch einmal erwähnt, dass die Versuchsträger eine Masse von 2,4 t (AUV DeepC) bzw. 1,7 t (AUV M63) besitzen. Schnellen Kurs- und Geschwindigkeitsvorgaben können die Fahrzeuge nur verzögert folgen, was sich in einem entsprechend großen Wenderadius bzw. einer langen Auslaufstrecke bei einem Bremsmanöver widerspiegelt. Die Ausweichmanöver müssen deshalb dieses Verhalten berücksichtigen und möglichst geringe Sollkursraten bzw. einen glatten Trajektorienverlauf besitzen. Eine Geschwindigkeitsverringerung bzw. ein Bremsmanöver, wie sie bei landgeführten Fahrzeugen zur Vermeidung einer Kollisionssituation eingesetzt werden, führt bei Unterwasserfahrzeugen nur begrenzt zum Erfolg und kann außerdem die Manövrierfähigkeit des Fahrzeuges beeinträchtigen. Deshalb wird diese Option bei den Ausweichmanövern nicht verwendet. Das Erreichen einer vorgegebenen Solltiefe erfolgt durch die Veränderung der Fahrzeugnicklage und ist damit immer mit einer horizontalen Positionsveränderung verbunden. Dies muss bei den vertikalen Ausweichmanövern berücksichtigt werden. Die Führung des Fahrzeuges während des Ausweichens muss auch bei einer Seeströmung sicher funktionieren. So sollen der vorgegebene Mindestabstand zu den Objekten eingehalten und Gebiete mit einer starken Gegenströmung gemieden werden.

### 3.2.4 Ausweichen in der vertikalen Ebene

Eine weitere Problematik bei Unterwasserfahrzeugen ist die Vermeidung eines möglichen Bodenkontaktes und die Kollision mit Erhebungen (Riffe, Steilwände, Deiche, Kaimauern). In [35] wird hierzu ein Kollisionsvermeidungssystem für das AUV REMUS ausschließlich für die vertikale Ebene vorgestellt. Dieses AUV wurde für den Flachwasserbereich zur Durchführung hydrografischer Analysen entwickelt. Ein vorausschauendes Sektor Scanning Sonar (engl.: forward-look sonar) bestimmt die Höhe und die Entfernung zu den in Fahrtrichtung liegenden Objekten, auf dessen Basis eine Anstiegsbahn berechnet wird, der das Fahrzeug durch Veränderung seiner Nicklage folgt.

Das Führungskonzept, in dem das Modul *FIS* implementiert ist, verwendet zur Vermeidung einer Bodenkollision ein zweistufiges Konzept. In der ersten Stufe kann während einer Mission als Tiefenvorgabe für die einzelnen Basismanöver neben einer *absoluten Tiefe* auch eine *Höhe über Grund* vorgegeben werden. Dabei wird das Fahrzeug beim Abfahren einer Mission immer in einem konstanten Abstand zum Meeresboden geführt. Abbildung 3.2 zeigt die Arbeitsweise der beiden Tiefenmodi. Voraus liegende Erhebungen bzw. Wände werden als Objekte an das Kollisionsvermeidungssystem gemeldet.

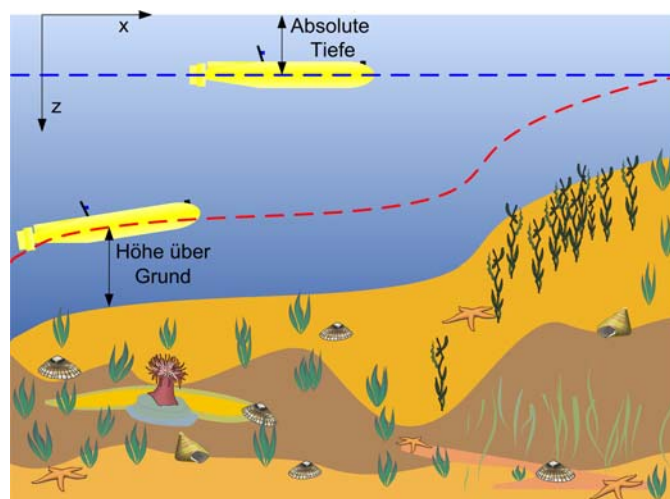


Abbildung 3.2: Tiefenmodi

Wird eine Kollisionssituation zwischen einem Basismanöver des Missionsplanes und einem Objekt im Submodul *Kollisionsüberwachung* ermittelt, wird in der zweiten Stufe das *Kollisionsvermeidungssystem* aktiviert. Dies würde passieren, wenn im Tiefenmodus *Höhe über Grund* das Bodenprofil so steil ist, dass das Fahrzeug seine Tiefe nicht schnell genug dem Profil anpassen kann und so zu dicht an den Meeresboden fährt. Im Tiefenmodus *Absolute Tiefe* liegt eine Kollisionssituation vor, wenn ein Basismanöver ein Objekt schneidet, das heißt ein Manöver wurde durch eine Meeresbodenerhebung geplant.

In der aktuellen Spezifikation/Arbeitsweise des *Autopiloten* wird im Tiefenmodus *Höhe über Grund* nur die Echolotinformation zur Bestimmung des Bodenabstandes für einen Tiefenregler verwendet. Ein vorausschauendes Handeln ist so nicht möglich. Eine Verbesserung wäre die Verwendung eines generierten Bodenprofils zur Bildung einer abzufahrenden vertikalen Sollbahn in Analogie zu [35]. Dabei wird das Nicklagekommando zur Beseitigung der vertikalen Bahnablage in Analogie zum Gierwinkelkommando bei der horizontalen Bahnregelung verwendet. Das Bodenprofil kann durch eine Datenfusion der Echolotininformation, der vertikalen Sonarinformationen sowie aus digitalen Seekarten generiert werden. Die gebildete vertikale Sollbahn berücksichtigt die maximal mögliche Nicklage des Fahrzeuges, wodurch rechtzeitig mit dem Auftauchvorgang vor einer Erhebung begonnen werden kann. Ist der Anstieg zu steil, kann der *Autopilot* das Fahrzeug im Schweben-Modus (engl.: hovering) über die Erhöhung führen. Diese vorausschauende Arbeitsweise des Autopiloten würde dafür sorgen, dass die erste Stufe möglichst lange zur Vermeidung einer Meeresbodenkollision aktiv ist und das Fahrzeug seine vorgegebene Mission durch Abfahren der Manöver erfüllen kann.

Wird das *Kollisionsvermeidungssystem* aktiviert, soll ein Ausweichen in der vertikalen Ebene durch Vorgabe von Tiefenwerten an den *Autopiloten* möglich sein. Dies ist die einzige Führungsgröße zur Veränderung der vertikalen Fahrzeugposition (siehe auch Abschnitt 8.2.1). Ein vertikales Ausweichmanöver ist an die beschränkte vertikale Manövrierfähigkeit des Fahrzeuges gebunden und hat nur Sinn, wenn eine genaue Kenntnis der Objektausdehnungen bekannt ist, um es so einem horizontalen Ausweichmanöver vorzuziehen. Deshalb soll ein vertikales Ausweichen nur bei dem planenden Ansatz möglich sein. Bei bekannten Objekten wie Kaimauern, Dämmen oder Erhebungen, die bis zur Wasseroberfläche reichen, wird durch die Vorgabe einer entsprechenden Zylinderhöhe bei den Objekten ein vertikales Ausweichmanöver bei der Planung ausgeschlossen.

### 3.2.5 Wegroute

Die generierte Wegroute soll zeit- bzw. wegoptimal sein und die Manövrierfähigkeit des Fahrzeuges berücksichtigen. Die Zeit zur Generierung einer Route sollte innerhalb eines Sonar-Updates von 4,5 s liegen. Eine Überprüfung der Befahrbarkeit der Route kann anhand einer *Digitalen Seekarte* durchgeführt werden. So können Kollisionen mit dem Meeresboden schon in der planenden Phase erkannt und durch entsprechende Planmodifikationen verhindert werden.

### 3.2.6 Anforderungskatalog

Nachfolgend sind noch einmal die wichtigsten Anforderungen an das *Kollisionsvermeidungssystem* aufgeführt:

- Das Kollisionsvermeidungssystem soll in Echtzeit abgearbeitet werden.
- Die detektierten Objekte werden als elliptische Zylinder übergeben. Sie sind hauptsächlich stationär und können erst spät detektiert werden.
- Es sind maximal 50 Objekte in einem vorgegebenen Operationshorizont von 200 m Länge, 100 m Breite und 100 m Höhe bei der Generierung der Ausweichmanöver einzubeziehen.
- Die beschränkte Manövrierfähigkeit des Fahrzeuges ist bei der Erzeugung der Ausweichmanöver zu berücksichtigen (Geringe Sollkursraten bzw. glatte Solltrajektorien sollen erzeugt werden. Die vertikalen Manöver sind auf die maximale Fahrzeugnicklage zu begrenzen.).
- Die Ausweichmanöver sollen auch bei einer vorhandenen Seeströmung sicher funktionieren.
- Die generierten Fahrtrajektorien sollen wegoptimal sein.
- Ein Ausweichen in der horizontalen und in der vertikalen Ebene soll möglich sein.

## 3.3 Architektur des Kollisionsvermeidungssystems

Auf Basis des im Abschnitt 3.2.6 definierten Anforderungskataloges und der im Abschnitt 3.1.2 vorgestellten Merkmale des reaktiven und des planenden Ansatzes wurde für das neu entwickelte *Kollisionsvermeidungssystem* eine hybride Struktur favorisiert. Dabei arbeiten ein planender und ein reaktiver Ansatz parallel. Dies steht im Widerspruch zu den vielfach hierarchisch aufgebauten Kollisionsvermeidungssystemen [71, 101]. Die Gründe für ein solches Vorgehen sollen deshalb nachfolgend erläutert werden.

Ändert sich die Umweltsituation durch neu detektierte Objekte oder durch einen verschobenen Zielpunkt, können bei einem hierarchischen Konzept die von der *Wegeplanung* an die *Reaktive Steuerung* übergebenen Anweisungen ihre Gültigkeit verlieren. So kann der übergebene Streckenabschnitt oder der Zwischenzielpunkt, der auf der Basis der alten Objektinformationen des Operationsgebietes generiert wurde, jetzt in einem Objekt liegen. Eine gezielte Führung des Fahrzeuges durch die *Reaktive Steuerung* wäre bis zur Bereitstellung neuer Vorgaben nach einer Neuplanung in der *Wegeplanung* nicht möglich. Dieses Problem besteht auch nach dem Start des Kollisionsvermeidungssystems, bei dem es noch keine Vorgaben für die *Reaktive Steuerung* gibt.

Aus diesem Grund arbeitet die *Reaktive Steuerung* als gleichgestelltes System, das die selben Zielpunktvorgaben und Hindernisinformationen, wie die *Wegeplanung* verwendet. So wird bei einer notwendigen Neuplanung des Weges die Führung an die *Reaktive Steuerung* übergeben. In Abbildung 3.3 ist das Flussdiagramm der Update-Routine des Kollisionsvermeidungssystems dargestellt. Zuerst wird in der *Wegeplanung* überprüft, ob ein neuer Manöverplan benötigt wird. Dieser ist erforderlich:

- nach dem Start des Kollisionsvermeidungssystems,
- nach einer Zielpunktverschiebung durch neue detektierte Hindernisse oder durch eine neue Manöverliste,
- wenn die aktuelle Bahnablage einen vorgegebenen Wert überschreitet,
- wenn Hindernisse mit dem bestehenden Weg kollidieren oder
- nach Ablauf einer vorgegebenen Zeit (So kann ggf. ein kürzerer Weg gefunden werden, wenn sich die Hinderniskonstellation seit der Plangenerierung geändert hat).

Muss ein neuer Manöverplan erstellt werden, wird die Wegerzeugung initiiert, welche in einem eigenständigen Thread läuft. So kann sie unabhängig von den anderen sequentiell abgearbeiteten Submodulen einen Plan generieren, ohne dabei die Update-Routine des Moduls *FIS* zu blockieren. Steht noch kein neuer Manöverplan zur Abarbeitung zur Verfügung, arbeitet während dieser Zeit die *Reaktive Steuerung* und generiert die Führungskommandos für das Fahrzeug. In der *Wegeplanung* werden die einzelnen Manöver des Manöverplanes sequentiell abgearbeitet. Dabei wird die Bahnablage  $e_{Ablage}$  und die Bahnrichtung  $\psi_{Bahn}$  des aktuellen Manövers bestimmt und an die Führungsalgorithmen übergeben (siehe Abbildung 3.4 und auch Abschnitt 8.3.3). Treten beim Abfahren des generierten Weges unvorhersehbare, plötzlich auftauchende Hindernisse auf, wird ein neuer Plan bei der Wegerzeugung angefordert und inzwischen die Fahrzeugführung an die *Reaktive Steuerung* übergeben. Die *Reaktive Steuerung* reagiert auf die im Nahbereich des Sonars aufgefassten Hindernisse durch entsprechende reaktive Steuerkommandos, wodurch ein wegoptimales Umfahren nicht immer möglich ist. Eine solche Forderung hat hier auch nicht oberste Priorität. Aufgabe ist ein sicheres und schnelles Ausweichen zur Vermeidung einer Kollisionssituation, wobei das Ausweichen nur in der horizontalen Ebene erfolgt.

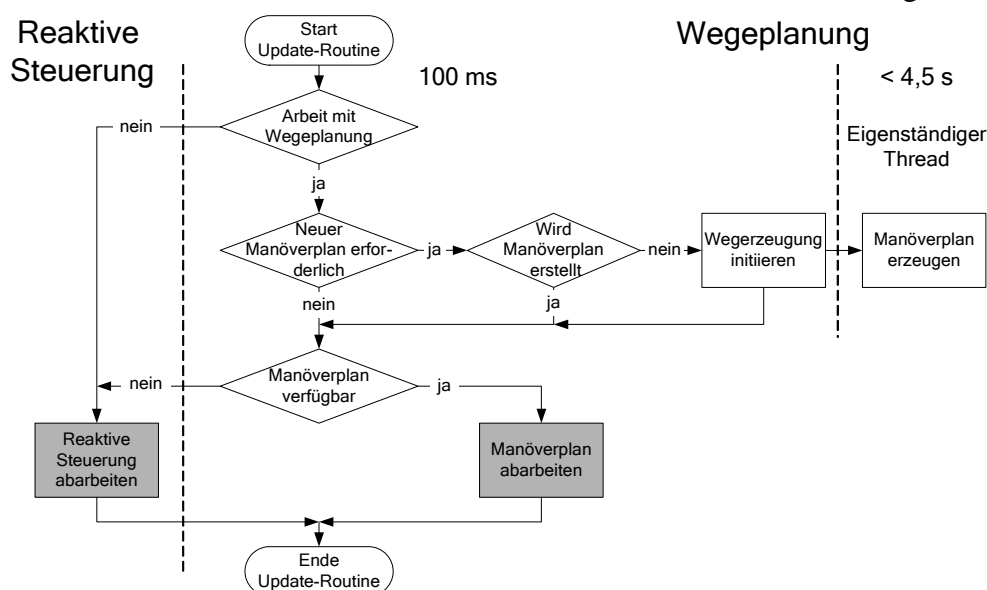


Abbildung 3.3: Flussdiagramm der Update-Routine des Kollisionsvermeidungssystems

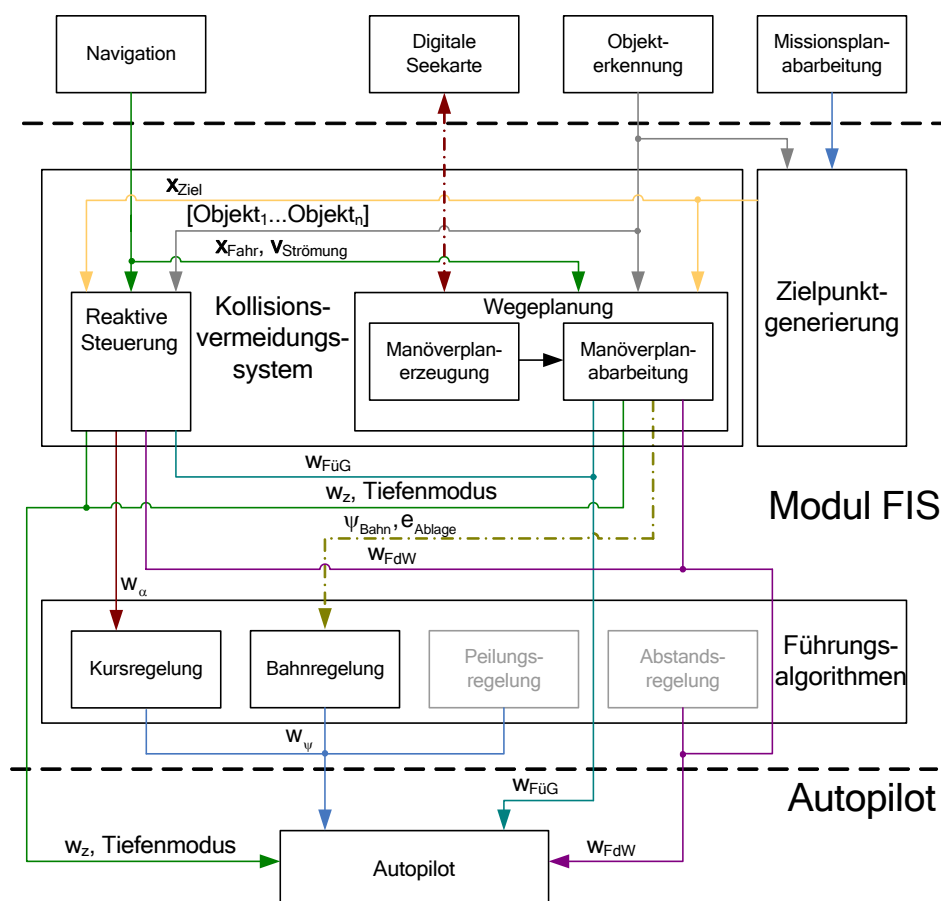


Abbildung 3.4: Struktur des Kollisionsvermeidungssystems

## 3.4 Zusammenfassung

Die Aufgabe des Submoduls *Kollisionsvermeidungssystem* besteht in der Führung des Fahrzeuges zu einem vorgegebenen Rendezvouspunkt auf dem Missionsplan unter Vermeidung von Hinderniskollisionen. Die generierten Ausweichmanöver berücksichtigen die Manövrierfähigkeit des Fahrzeuges, eine vorhandene Seeströmung sowie die Forderung nach einem wegoptimalen Verlauf.

Im einführenden Teil dieses Kapitels erfolgt ein Vergleich des reaktiven und des planenden Ansatzes zur Kollisionsvermeidung. Anhand ausgewählter Architekturen von Kollisionsvermeidungssystemen werden unterschiedliche Lösungsansätze vorgestellt und bewertet. Eine detaillierte Anforderungsbeschreibung an das Kollisionsvermeidungssystem zeigt die zu lösenden Probleme im Unterwasserbereich auf. Wesentliche Probleme sind die begrenzte Manövrierfähigkeit des Fahrzeuges, das Fahren bei einer vorhandenen Seeströmung sowie die verzögerte Bereitstellung der Hindernisinformationen.

Das neu entwickelte *Kollisionsvermeidungssystem* besitzt eine hybride Architektur. Dabei arbeiten ein reaktiver und ein planender Ansatz parallel. Die *Reaktive Steuerung* übernimmt die Führung des Fahrzeuges, während ein neuer Manöverplan in der *Wegeplanung* generiert wird. Steht ein Manöverplan dann zur Verfügung, arbeitet die *Wegeplanung* diesen ab. Kommt es zu einem zu starken Variieren der Objektdaten, wird nur mit der *Reaktiven Steuerung* gearbeitet. Der Test und die Bewertung des neu entwickelten Kollisionsvermeidungssystems erfolgen im Kapitel 9.

## 4 Wegeplanung

Die *Wegeplanung* stellt den planenden Teil des Kollisionsvermeidungssystems dar. Sie umfasst eine *planende* und eine *ausführende* Funktion. Bei der *planenden* Funktion werden die gesamten Informationen über das aktuelle Operationsgebiet zur Generierung eines Manöverplanes verwendet. Das sind neben den Informationen des Sonars, die gesammelten Hindernisdaten der bisherigen Mission sowie die Daten einer *Digitalen Seekarte*. Bei der Entwicklung des Wegeplanungsalgorithmus wurde der Schwerpunkt auf die Erstellung eines optimalen Routenplanes innerhalb der einzuhaltenden Zeitvorgabe gelegt. Zur Erfüllung dieser Anforderungen wurden graphenbasierte Ansätze favorisiert. Nach der Erstellung eines Manöverplanes erfolgt seine Abarbeitung durch die *ausführende* Funktion der *Wegeplanung*. Dabei werden die einzelnen Manöver des generierten Manöverplanes sequentiell abgearbeitet und die entsprechenden Steuerkommandos zur Bahnführung des Fahrzeuges an das Submodul *Führungsalgorithmen* übergeben.

Die einzelnen Schritte, von der Erzeugung eines geometrischen Graphen, über die Suche nach dem kürzesten Weg, der anschließenden Generierung eines abzufahrenden Manöverplanes sowie dessen Abarbeitung, werden in diesem Kapitel behandelt. Zwei entwickelte Verfahren zur Generierung eines geometrischen Graphen aus einem Konfigurationsraum mit elliptischen Objektzylindern sowie ein Algorithmus zur Bestimmung der Fahrtkosten unter Einbeziehung der Strömungsinformation werden detailliert beschrieben. Abschließend erfolgt ein Vergleich der zwei Verfahren anhand durchgeführter Laufzeitüberprüfungen und der generierten Routenverläufe.

### 4.1 Einführung in die Wegeplanung

#### 4.1.1 Bekannte Ansätze

Zur Erzeugung eines Routenplanes gibt es eine Vielzahl von Lösungsmöglichkeiten aus den Gebieten der Mathematik, der Optimierung und der Informatik. Nachfolgend werden einige dieser Verfahren vorgestellt. So erhält der Leser einen Eindruck von der unterschiedlichen Herangehensweise bei der Lösung dieser Aufgabe.

In [152] wird eine Evolutionsstrategie zur Ermittlung eines zeitoptimalen Weges für einen mobilen Roboter eingesetzt. Der Weg wird durch einen Vektor aus Steuerwinkeln beschrieben, welche der Reihe nach und zu diskreten Zeitpunkten an den Roboter übergeben werden. So wird der mit einer konstanten Geschwindigkeit fahrende Roboter anhand der vorgegebenen Richtungswechsel durch den Hindernisparcours geführt. Die Optimierungsaufgabe besteht in der Ermittlung eines solchen Steuerwinkelvektors, der den Roboter in einer minimalen Zeit kollisionsfrei durch den Hindernisparcours führt. Die zu optimierenden

Parameter sind die einzelnen Steuerwinkel des Vektors und dessen Größe. Diese Größe korreliert mit der benötigten Fahrzeit. So werden an die einzelnen Individuen der Startpopulation Parametervektoren mit unterschiedlicher Größe übergeben. Durch die Evolutionsmechanismen des Crossovers und der Mutation können während der Optimierung Teile der Parametervektoren getauscht bzw. ihre Parameter modifiziert werden, mit dem Ziel, einen Weg durch den Hindernisparcours in der kürzesten Zeit zu finden.

In [130] werden Neuronale Netze zur Modellierung, Positionsregelung und Wegeplanung für das AUV Twin Burger 2 eingesetzt. Zum Trainieren des Netzes für die Wegeplanung wird das Reinforcement Learning (bestärkendes Lernen) benutzt. Dazu besteht die neuronale Netzwerkarchitektur aus zwei Neuronalen Netzen, einer Vorhersage- und einer Lerneinheit.

Zur Bestimmung energieoptimaler Trajektorien für ein Unterwasserfahrzeug wurde in [139] das Verfahren der Optimalen Steuerung eingesetzt. Dabei wird eine numerische Lösung eines optimalen Steuerungsproblems ermittelt. Dies erfolgt unter Einbeziehung der Fahrzeugdynamik, der Propellercharakteristik sowie des Energieverbrauches. Die Hindernisform wird durch einen Kreis nachgebildet, so dass die Kreisgleichung als Beschränkung in der Form:

$$c(\mathbf{x}) = (x - x_{\text{Kreis}})^2 + (y - y_{\text{Kreis}})^2 - r^2 \geq 0 \quad (4.1)$$

mit in das optimale Steuerungsproblem einbezogen wird. Die Verwendung der Kreisgleichung zur Beschreibung der Hindernisse wird auch in [150] zur Aufstellung von Ungleichungsbeschränkungen eingesetzt. Der Weg wird dort durch eine Optimierung mit Ungleichungsnebenbedingungen ermittelt. Dabei wird eine parabolische Gütefunktion vorgegeben, deren Minimum die Zielposition darstellt. Der Weg ergibt sich durch Verfolgen des Iterationsverlaufes von der Startposition aus bis zum Erreichen des Minimums.

Die leistungsfähigsten Verfahren zur Bestimmung eines Routenplanes sind die graphenbasierten Ansätze. Dabei wird ein Routenplan unter Verwendung eines geometrischen Graphen ermittelt. Zur Erstellung eines solchen Graphen werden Punkte (Knoten) im Operationsgebiet definiert, die vom Fahrzeug befahrbar sind. Die befahrbaren Verbindungen zwischen diesen Punkten werden als Kanten in den Graphen eingetragen. Jede Kante besitzt eine Bewertung (Kosten, Gewicht), welche die Länge der Verbindung, die entstehenden Kosten beim Abfahren der Verbindung oder die dafür benötigte Zeit sein kann (siehe Abbildung 4.1). Nach Generierung des Graphen wird ein Weg (Routenplan) vom Anfangsknoten (Startpunkt) zum Endknoten (Zielpunkt) mit den geringsten Gesamtkosten durch einen Suchalgorithmus ermittelt. Dieser durchsucht den Graphen, um eine Kombination von Kanten zu ermitteln, die den Startknoten und den Endknoten mit der geringsten Summe an Kosten verbinden.

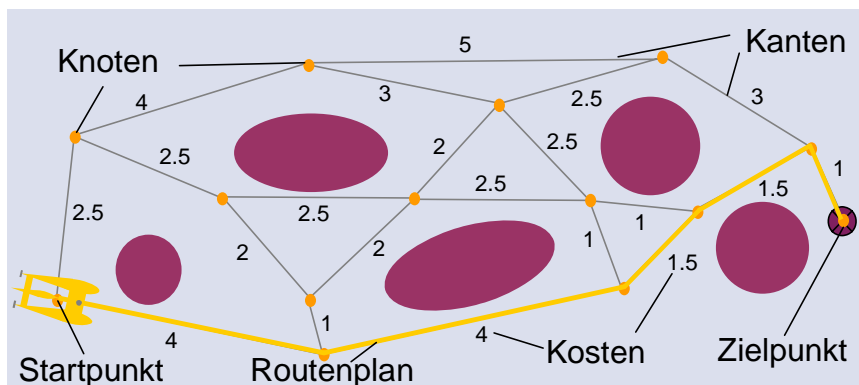


Abbildung 4.1: Geometrischer Graph



## 4.1.2 Darstellungsmöglichkeiten der Umwelt in einem Graphen

### 4.1.2.1 Gitterbasierte Darstellung

Die einfachste Möglichkeit der Graphenerzeugung besteht in der Zerlegung des Operationsgebietes in eine gleichmäßige Gitterstruktur, wie sie in Abbildung 4.2 dargestellt ist. Der Vorteil dieses Verfahrens besteht in der einfachen Bestimmung der Sektoren und ihrer Repräsentation in einer Matrixstruktur. Aufgrund dieser gleichmäßigen Struktur gestaltet sich die Generierung des Graphen sehr einfach. Die so entstandenen rechteckigen Sektoren haben alle die gleichen Abmaße, wodurch die Repräsentationsgüte des Operationsgebietes direkt von ihrer Größe abhängt. Eine gute Beschreibung der an die Hindernisse angrenzenden *freien* Sektoren ist jedoch nur mit einer hinreichend fein gewählten Sektorgröße möglich. Sonst besteht die Gefahr, dass befahrbare Gassen zwischen den Hindernissen für eine Durchfahrt gesperrt bleiben. Des Weiteren ist durch eine zu grobe Sektorgröße ein optimales Umfahren der Hindernisse nicht gegeben. Wählt man die Sektorgröße fein genug, werden die oben beschriebenen Nachteile vermieden, es kommt aber zu einer Erhöhung des Speicherbedarfes.

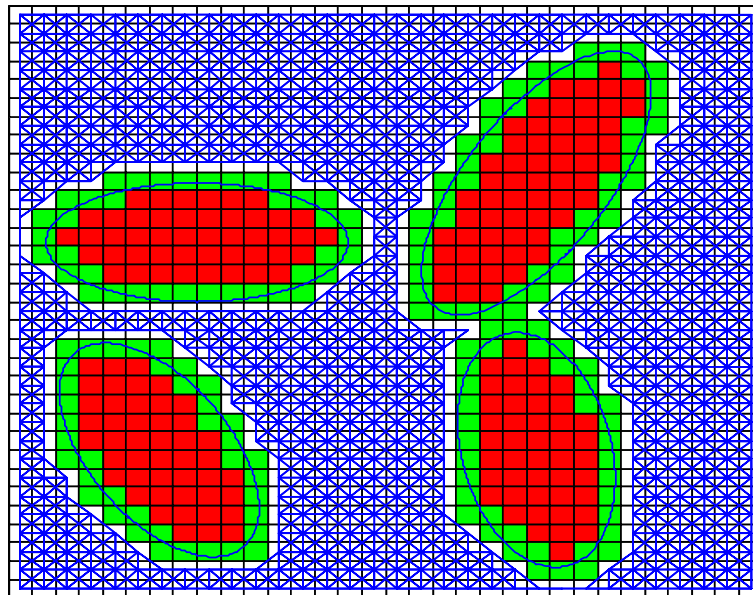


Abbildung 4.2: Zerteilung in gleich große Sektoren

### 4.1.2.2 Quadtree- / Octreegraph

Bei diesem Verfahren wird das Operationsgebiet in quadratische (2D-Raum) bzw. kubische (3D-Raum) Sektoren unterteilt. In Abhängigkeit von darin vorhandenen Hindernissen können sie den Zustand *frei*, *belegt* oder *teilweise belegt* besitzen. *Teilweise belegte* Sektoren werden rekursiv weiter geteilt, bis eine maximale Teilungstiefe bzw. eine minimale Sektorgröße erreicht wurde. Als *frei* oder *belegt* klassifizierte Sektoren werden nicht weiter geteilt. So findet eine systematische Zerlegung des Operationsgebietes, analog der Verästelung eines Baumes, statt. Nach Erzeugung aller Sektoren wird der Graph aufgebaut. Dazu bilden die Mittelpunkte aller hindernisfreien Sektoren die Knoten des Graphen. Die Kanten sind die Verbindungen zwischen benachbarten hindernisfreien Sektoren (siehe Abbildung 4.3). Gegenüber der gitterbasierten Darstellung kann mit einem Quadtreegraphen eine detailliertere Unterteilung des Operationsgebietes in *freie* und *belegte* Sektoren bei geringerem Speicheraufwand erfolgen.

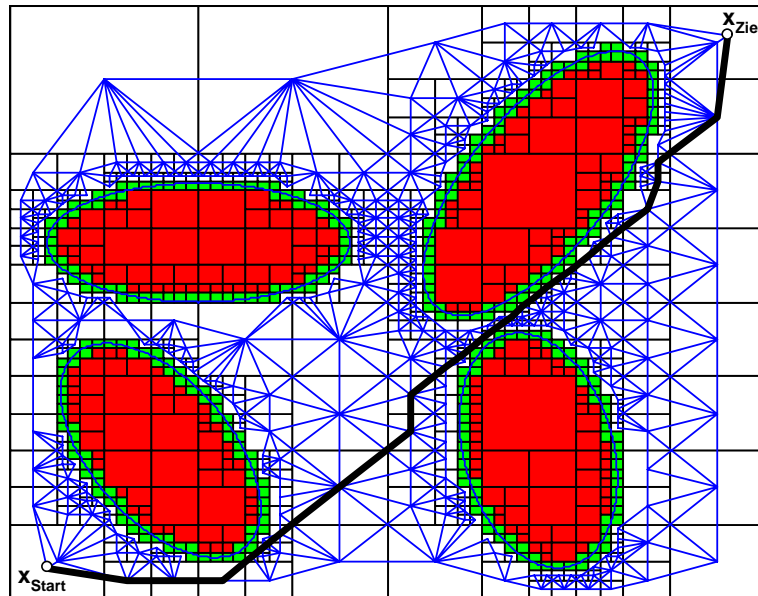


Abbildung 4.3: Quadtreegraph

### 4.1.2.3 Sichtbarkeitsgraph

Beim Sichtbarkeitsgraphen bilden die Eckpunkte der Hindernisse die Knoten des Graphen. Hindernisse ohne Ecken, wie Kreise oder Ellipsen, werden durch eine Polygonform nachgebildet, die das Hindernis vollständig einschließt. Alle möglichen Linien zwischen den Knoten, die keine Hindernisse schneiden, werden zu Kanten des Graphen. Abbildung 4.4 zeigt einen so generierten Graphen mit der in dieser Arbeit verwendeten elliptischen Hindernisform. Der ermittelte Weg bei Verwendung eines Sichtbarkeitsgraphen führt immer über die Hindernisecken zum Zielpunkt. Deshalb ist es bei diesem Verfahren wichtig, die Hindernisse um einen vorgegebenen Sicherheitsabstand zu vergrößern, um ein Kollidieren des Fahrzeuges mit den Hindernissen zu vermeiden.

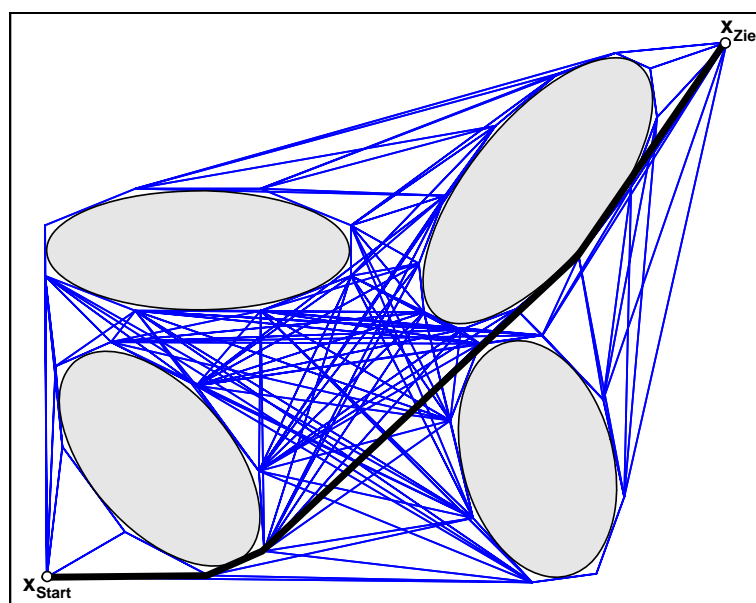


Abbildung 4.4: Sichtbarkeitsgraph

#### 4.1.2.4 Kreistangentenansatz

Für die Wegeplanung des mobilen Roboters *Stanford Cart* wurden die im Operationsgebiet vorhandenen Hindernisse durch Kreise modelliert [115]. Dabei existieren vier Wege zwischen je zwei Hindernissen, welche sich aus den Tangenten ihrer Kreise ergeben. Diese Tangenten sowie die zu umfahrenden Kreisbögen bilden die möglichen Kanten eines Graphen, ihre Endpunkte die dazugehörigen Knoten. Nach einer erfolgreichen Kollisionsüberprüfung der Kanten mit allen im Operationsgebiet vorhandenen Hindernissen werden sie mit den dazugehörigen Knoten in den Graphen eingetragen. Diese Art der Kantenbestimmung wird mit allen möglichen Hindernispaaren des Operationsgebietes durchgeführt.

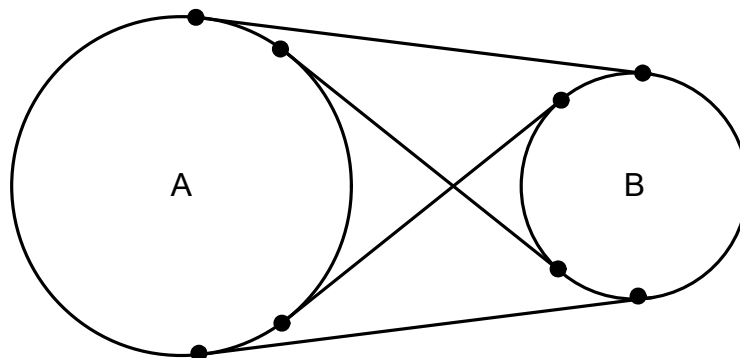


Abbildung 4.5: Mögliche Wege zwischen Hindernis A und B

#### 4.1.3 Suchalgorithmen für kürzeste Wege in einem Graphen

Zur Ermittlung eines Weges zwischen einem gegebenen Anfangsknoten und einem Endknoten gibt es in der Graphentheorie viele Verfahren. Zu den einfachsten Verfahren gehört die Breitensuche (engl. Breadth-First). Dabei wird der Graph vom Anfangsknoten ausgehend in die Breite nach einem vorgegebenen Knoten durchsucht [155]. Dies erfolgt dadurch, dass man jeden Knoten, der über eine Kante mit dem Anfangsknoten verbunden ist, aufsucht und überprüft, ob er schon aufgesucht wurde oder der gesuchte Knoten ist. Wurde der Knoten noch nicht aufgesucht, wird er am Ende einer Liste eingefügt. Wenn alle Knoten besucht wurden, wird im nächsten Schritt der erste Knoten aus der Liste entnommen und das Verfahren in oben beschriebener Weise wiederholt. Dies wird dann für jeden in der Liste enthaltenen Knoten durchgeführt. So wird Ebene für Ebene überprüft, bis der gesuchte Knoten gefunden wurde. Der so ermittelte Weg besteht aus der geringsten Anzahl an Kanten-elementen bzw. der Anzahl an durchlaufenen Ebenen, weshalb er nicht gleichzeitig auch der Kürzeste sein muss. So liefert dieses Verfahren nur eine suboptimale Lösung, da die Kantengewichte bei der Suche nicht einbezogen werden. Der von Dijkstra entwickelte Algorithmus berechnet einen Weg mit den geringsten Kosten in einem kantengewichteten Graphen [41]. Dabei wird der Graph ausgehend vom Anfangsknoten vollständig untersucht, um den Weg mit den geringsten Kosten zum Endknoten zu finden. Der Algorithmus liefert als Ergebnis immer den Weg mit den geringsten Kosten. Der auf den Dijkstra-Algorithmus aufbauende A\*-Algorithmus verwendet zusätzlich eine Heuristik, um zielgerichteter nach dem Endknoten zu suchen [117]. Dabei werden zuerst die Knoten mit den geringsten geschätzten Kosten zum Endknoten in die Suche einbezogen. Diese Strategie führt zu einer beschleunigten Abarbeitung gegenüber dem Dijkstra-Algorithmus. Durch die verwendete Heuristik kann aber nicht sichergestellt werden, dass immer der kürzeste Weg gefunden wird.

## 4.2 Beschreibung der Wegeplanung

### 4.2.1 Anforderungen

Die Forderungen an die *Wegeplanung* folgen aus den in Abschnitt 3.2 definierten Anforderungen an das *Kollisionsvermeidungssystem*. Nachfolgend sollen die Wichtigsten noch einmal dargelegt werden:

- Die Zeit zur Generierung eines Manöverplanes soll innerhalb eines Sonar-Updates von 4,5 s liegen.
- Die detektierten Objekte werden als elliptische Zylinder nachgebildet.
- Die erzeugten Manöverpläne sollen wegoptimal sein und die Strömungsinformation berücksichtigen.
- Es sind maximal 50 statische Objekte bei der Wegeplanung einzubeziehen.

### 4.2.2 Vorstellung des Konzeptes

Die Einbindung der *Wegeplanung* in das *Kollisionsvermeidungssystem* wurde ausführlich im Abschnitt 3.3 beschrieben. In diesem Abschnitt sollen die zu lösenden Aufgaben des Moduls *Wegeplanung* im Überblick vorgestellt werden.

Die Wegeplanung setzt sich aus einer *planenden* und einer *ausführenden* Komponente zusammen. Die *ausführende* Komponente arbeitet innerhalb der Update-Routine des Moduls *FIS* den in der *planenden* Komponente erstellten Manöverplan ab. Ein solcher Manöverplan besteht aus Strecken- und Kreisbogenmanövern, wie sie in Abschnitt 1.3.3 beschrieben wurden. Die Algorithmen zur Abarbeitung eines Manöverplanes werden detailliert in Abschnitt 4.6 vorgestellt. Wird ein Manöverplan abgearbeitet, erfolgt mit jedem Empfang der Objektdaten seine Kollisionsüberprüfung mit diesen Objekten. Findet eine Kollision statt, wird der bestehende Plan verworfen und eine Neuplanung initiiert. Eine solche Neuplanung erfolgt auch, wenn die aktuelle Bahnablage einen vorgegebenen Wert überschreitet. Weitere Gründe für eine Neuplanung sind in Abschnitt 3.3 aufgeführt. Die Erzeugung eines neuen Manöverplanes erfolgt in einem eigenständigen Thread, so dass es nicht durch die rechenintensiven Algorithmen zu einer Blockierung in der *FIS* Update-Routine kommen kann, welche alle 100 ms abgearbeitet werden muss. Die einzelnen Schritte zur Erzeugung eines Manöverplanes sind in Abbildung 4.6 dargestellt. Aus den in Abschnitt 4.2.1 vorgestellten Anforderungen, einer umfangreichen Literaturrecherche sowie durchgeführten Untersuchungen zur Wegeplanung wurden Graphenverfahren zur Lösung der Wegplanungsaufgabe favorisiert. So besteht der erste Schritt in der Generierung eines geometrischen Graphen. In Abschnitt 4.3 werden zwei Verfahren zur Erzeugung eines solchen geometrischen Graphen vorgestellt. Die Auswahl dieser Verfahren erfolgte auf der Grundlage der vorgegebenen geometrischen Form zur Beschreibung der Objekte (elliptischer Zylinder) und der Forderung nach effizienten Algorithmen, die mit den zur Verfügung stehenden Prozessor- und Speicherressourcen auskommen. Des Weiteren sollten die Verfahren innerhalb eines Sonar-Updates einen generierten Graphen zur Verfügung stellen. Nach Erstellung des Graphen wird in der zweiten Stufe ein kürzester Weg durch den Graphen unter Verwendung des Dijkstra- oder des A\*-Algorithmus ermittelt. Aus dem gefundenen Routenplan wird nach seiner Glättung (siehe Abschnitt 4.4.2) ein Manöverplan erzeugt. Bei der Erstellung des

Manöverplanes werden Kreisbogenmanöver zwischen den einzelnen Streckenabschnitten des Routenplanes eingefügt, welche das Bewegungsverhalten des Fahrzeuges beim Einfahren auf das neue Streckenelement beschreiben. In der letzten Stufe wird der erzeugte Manöverplan auf Kollision mit den Objekten sowie auf Befahrbarkeit durch die *Digitale Seekarte* überprüft.

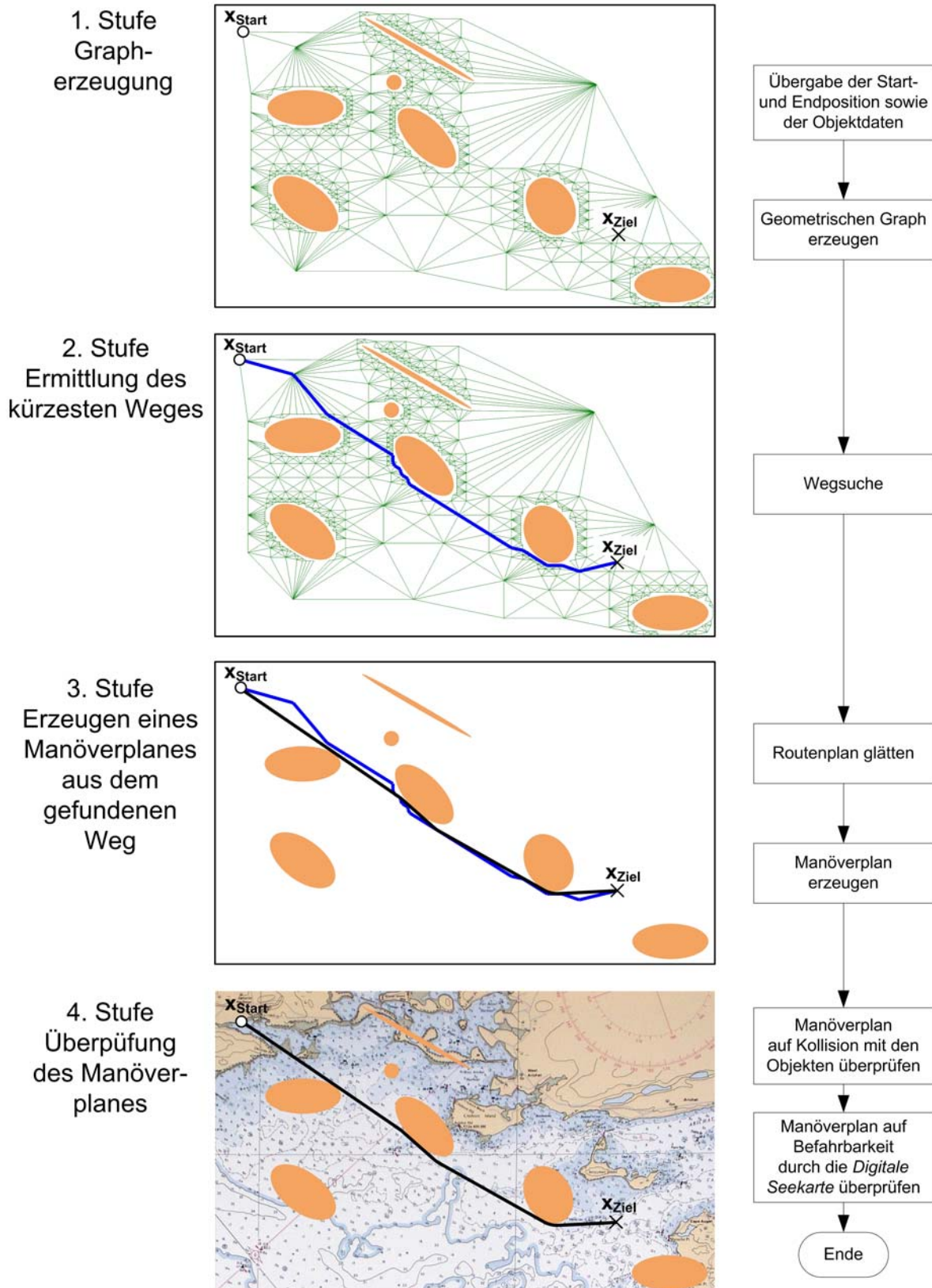


Abbildung 4.6: Die einzelnen Schritte zur Erzeugung eines Manöverplanes

## 4.3 Grapherzeugung

### 4.3.1 Quadtree- / Octreegraph

Das Verfahren zur Erzeugung eines Quadtreegraphen besteht aus zwei Phasen. In der ersten Phase findet eine hierarchische Zerteilung des Operationsgebietes in einzelne Sektoren statt. Diese Zerteilung erfolgt in Abhängigkeit von im Operationsgebiet vorhandenen Objekten und hat das Ziel, das Operationsgebiet möglichst genau in befahrbare und nicht befahrbare Sektoren aufzuteilen. Jeder Mittelpunkt eines befahrbaren Sektors bildet einen Knoten des zu erzeugenden Graphen. In einer zweiten Phase werden die Nachbarschaftsbeziehungen zwischen den einzelnen befahrbaren Sektoren ermittelt und benachbarte Knoten durch eine Kante verbunden. Diese zweistufige Vorgehensweise wird auch in anderen Arbeiten verwendet [101].

#### 4.3.1.1 Zerteilungsverfahren des Operationsgebietes

Die nachfolgenden Darlegungen werden im zweidimensionalen Raum für den Quadtreegraphen vorgestellt. Eine Erweiterung der Algorithmen auf den dreidimensionalen Raum ist jedoch leicht möglich. Die Zerteilung des Operationsgebietes in einzelne Sektoren erfolgt auf Basis eines hierarchischen Zerteilungsverfahrens [101]. Ausgehend vom Operationsgebiet, das den Ursektor darstellt, findet seine Zerteilung in separate Sektoren statt. Befinden sich in den entstandenen Sektoren Objekte, erfolgen weitere Zerteilungen dieser Sektoren. Ziel dieser wiederholten Zerteilungen ist es, das Operationsgebiet möglichst genau in *freie* Sektoren und von Objekten *belegte* Sektoren zu unterteilen. Bei der Zerteilung eines Sektors wird für jede Koordinatenrichtung ein Zerteilungsschritt durchgeführt, welcher den jeweiligen Sektor in zwei Teile halbiert. Die Zerteilung eines Sektors erfolgt in der Funktion *Sektorzerteilung(S)*, deren Ablauf in Abbildung 4.7 dargestellt ist. Eine Zerteilung wird durchgeführt, solange der Sektor teilweise mit Objekten belegt ist und die vorgegebene maximale Teilungstiefe  $t_{max}$  des Sektors noch nicht erreicht wurde.

Ist der Sektor vollkommen mit einem Objekt belegt, findet keine weitere Zerteilung des Sektors statt. Befinden sich keine Objekte im Sektor, werden die Informationen des Sektors in eine Baumstruktur eingetragen und die Funktion ohne Zerteilung beendet. Das Abspeichern der Informationen über freie Sektoren in eine Baumstruktur ist für die Ermittlung ihrer Nachbarschaftsbeziehungen in der zweiten Phase des Verfahrens notwendig. Eine detaillierte Beschreibung des Aufbaus und der Verwendung dieses Baumes erfolgt im Abschnitt 4.3.1.2. Bei jeder durchgeführten Teilung werden die Informationen über die Teilungsachse und die Teilungsposition (Splitgerade) als Knoten in die Baumstruktur eingetragen. Nach durchgeführter Zerteilung wird für die entstandenen Sektoren ebenfalls die Funktion *Sektorzerteilung(S)* aufgerufen, so dass eine rekursive Abarbeitung des Algorithmus entsteht.

Ein Sektorelement wird durch seine Mittelpunktposition  $x$  und den Belegungszustand (*frei*, *teilweise belegt*, *belegt*) beschrieben. Zur Durchführung des Zerteilungsalgorithmus umfasst ein Sektorelement noch zusätzliche Eigenschaften, wie die aktuelle Teilungstiefe  $t$ , eine Begrenzungsbox, einen Objektindex-Vektor  $v_{Oi}$  und einen Eckenbelegungs-Vektor  $v_{Eb}$  (siehe Abbildung 4.8). Eine achsengerichtete Begrenzungsbox beschreibt die Ausdehnung des Sektors. Diese Box wird durch zwei Punkte  $x_{min}$  und  $x_{max}$  definiert, welche die minimale und maximale Begrenzung des Sektors in den einzelnen Koordinatenachsen charakterisieren.

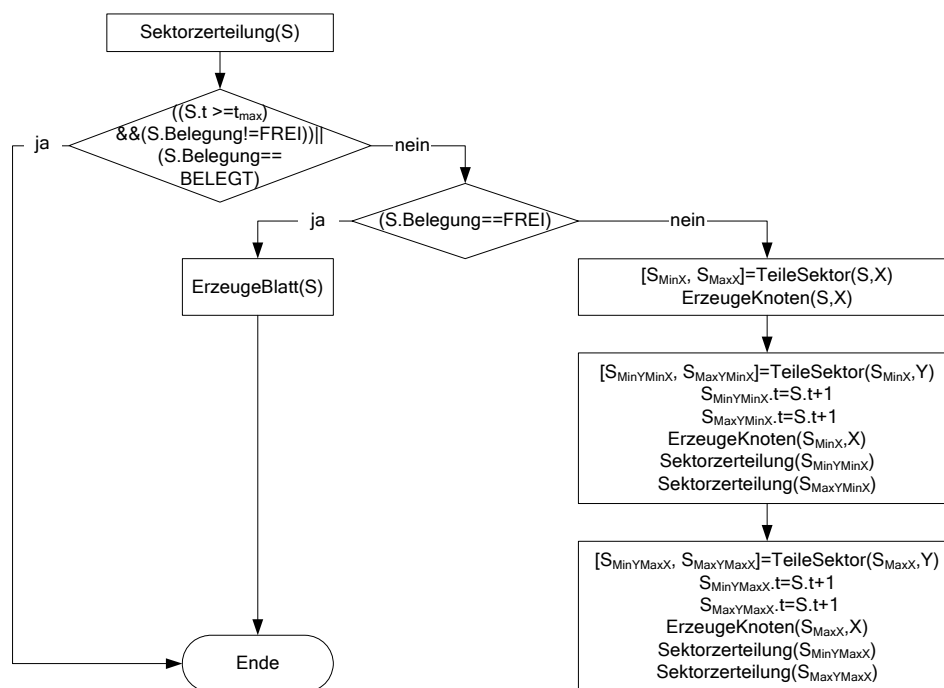


Abbildung 4.7: Flussdiagramm der Sektorzerteilung

Der Objektindex-Vektor umfasst die Nummern der Objekte, die sich innerhalb des Sektors befinden. Dabei müssen die Objekte nicht vollständig im Sektor liegen. Sobald sie eine Kante des Sektors schneiden, werden sie in den Objektindex-Vektor aufgenommen.

Um eine Aussage über den Belegungszustand eines Sektors mit Objekten treffen zu können, wurde der Eckenbelegungs-Vektor definiert. Jedes Element dieses Vektors ist einem im Sektor befindlichen Objekt zugeordnet und enthält eine Liste der Belegungszustände der einzelnen Sektorecken durch dieses Objekt. Erst wenn alle vier Sektorecken von ein und demselben Objekt überdeckt werden, gilt der Sektor als *belegt* (siehe Abbildung 4.9 Fall d). Eine solche Festlegung ist durch die konvexe Form der Objekte möglich. Ist der Objektindex-Vektor leer, hat der Sektor den Belegungszustand *frei* (siehe Abbildung 4.9 Fall a). Für den Fall, dass zwei oder mehrere Objekte durch ihr Zusammenwirken die Fläche des Sektors vollständig überdecken, gilt der Sektor immer noch als *teilweise belegt*. Dies führt jedoch nicht zu einem Fehler bei den weiteren Sektorzerteilungen der Nachfolger dieses Sektors. Die Sektorzerteilungen werden dann solange vorgesetzt, bis die Sektoren von einem Objekt vollkommen überdeckt werden oder die maximale Teilungstiefe erreicht ist. Der Belegungszustand der so gebildeten Sektoren ist entweder *teilweise belegt* oder *belegt*. Die korrekte Bestimmung des Belegungszustandes für diesen Sonderfall wäre sehr rechenaufwendig. Durch die vorgegebenen Objektkonstellationen tritt dieser Sonderfall jedoch sehr selten auf, weshalb auf eine korrekte Bestimmung des Belegungszustandes verzichtet wird.

Sektorelement S
Mittelpunktposition: $\mathbf{x}$
Belegung: {FREI, TEILWEISE BELEGT, BELEGT}
Teilungstiefe: $t$
Achsenorientierte Begrenzungsbox: $\mathbf{x}_{\min} \mathbf{x}_{\max}$
Objektindex-Vektor: $\mathbf{v}_{OI}$
Eckenbelegungs-Vektor: $\mathbf{v}_{Eb}$

Abbildung 4.8: Eigenschaften eines Sektorelements



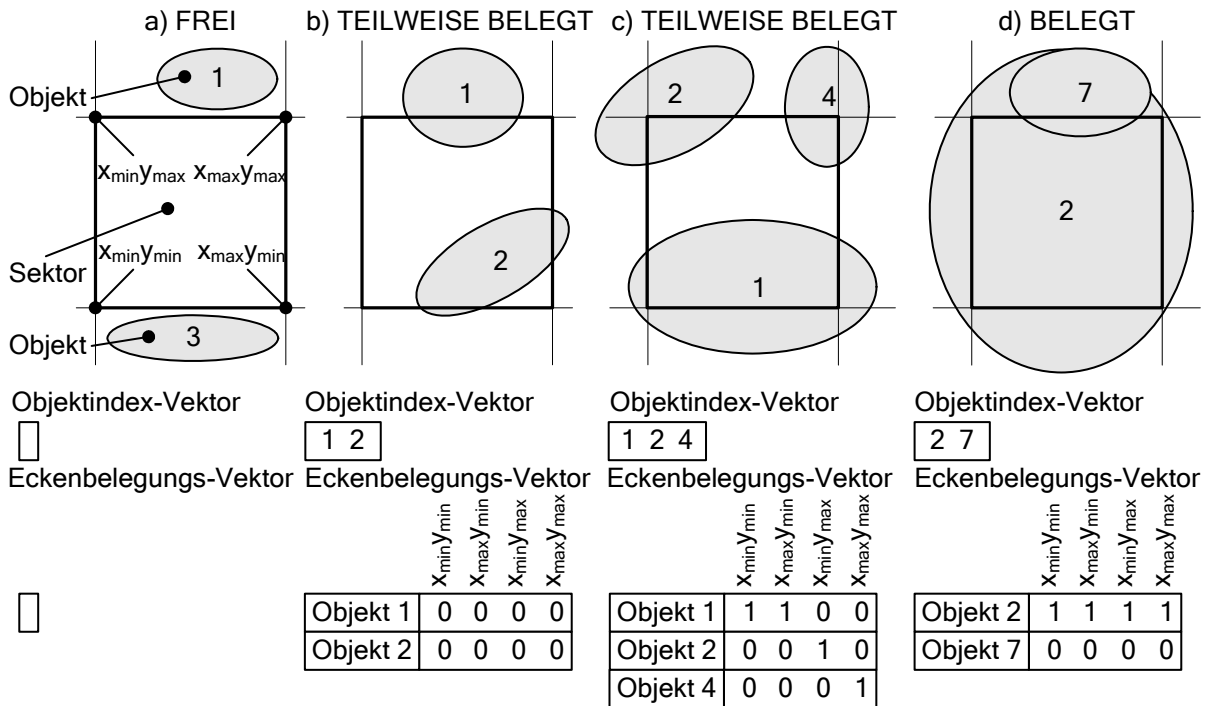


Abbildung 4.9: Bestimmung des Belegungszustandes des Sektors

Die Teilung eines Sektors erfolgt in der Funktion *TeileSektor*( $S, k$ ) (siehe Abbildung 4.7), die den Sektor entsprechend der vorgegebenen Splitkoordinate  $k$  in zwei Sektoren entlang der Splitgeraden halbiert und deren Eigenschaften bestimmt. Gestaltet sich die Bestimmung der Begrenzungsboxen und der Mittelpunktpositionen für die neuen Sektoren als trivial, ist zur Ermittlung ihrer Objektindex-Vektoren und Eckenbelegungs-Vektoren ein gewisser Rechenaufwand notwendig. Die beiden Vektoren werden durch separate Untersuchungen für jedes im zu teilenden Sektor befindliche Objekt sukzessiv erstellt. Abbildung 4.10 stellt die einzelnen Schritte des Algorithmus zur Teilung eines Sektors dar. Dabei werden die einzelnen Objekte laut Objektindex-Vektor des Sektorelements an die Funktion *BestimmeSektoreigenschaftenX* bzw. *BestimmeSektoreigenschaftenY* in Abhängigkeit der Splitkoordinate  $k$  übergeben. Die Abbildung A.2 im Anhang zeigt das Flussdiagramm zur Bestimmung der Eigenschaften der neu entstandenen Sektoren  $S_{min}$  und  $S_{max}$  für die x-Splitkoordinate. In dieser Funktion werden:

- durch Auswertung der Lage der Objektbegrenzungsbox zur Splitgeraden,
- durch Bestimmung und Auswertung der Schnittpunkte der Splitgeraden mit der elliptischen Form des Objektes,
- aus den Informationen über die Eckenbelegungszustände des zu teilenden Sektors
- und durch Bestimmung und Auswertung des Schnittpunktes der oberen bzw. unteren Kante des Sektors mit der elliptischen Form des Objektes

die Zugehörigkeit des Objektes zu den beiden neuen Sektoren sowie ihre Eckenbelegungszustände durch das Objekt ermittelt. Die Bestimmung der Eigenschaften für die y-Koordinate erfolgt in Analogie zur x-Koordinate, wobei lediglich die Koordinatenachsen vertauscht werden. Wird im dreidimensionalen Raum gearbeitet, gestaltet sich die Zuordnung der Objekte zu den neuen Sektoren in der z-Koordinate sehr einfach. Hier braucht nur die z-Ausdehnung des Objektes mit der Begrenzungsbox des Sektors verglichen zu werden.



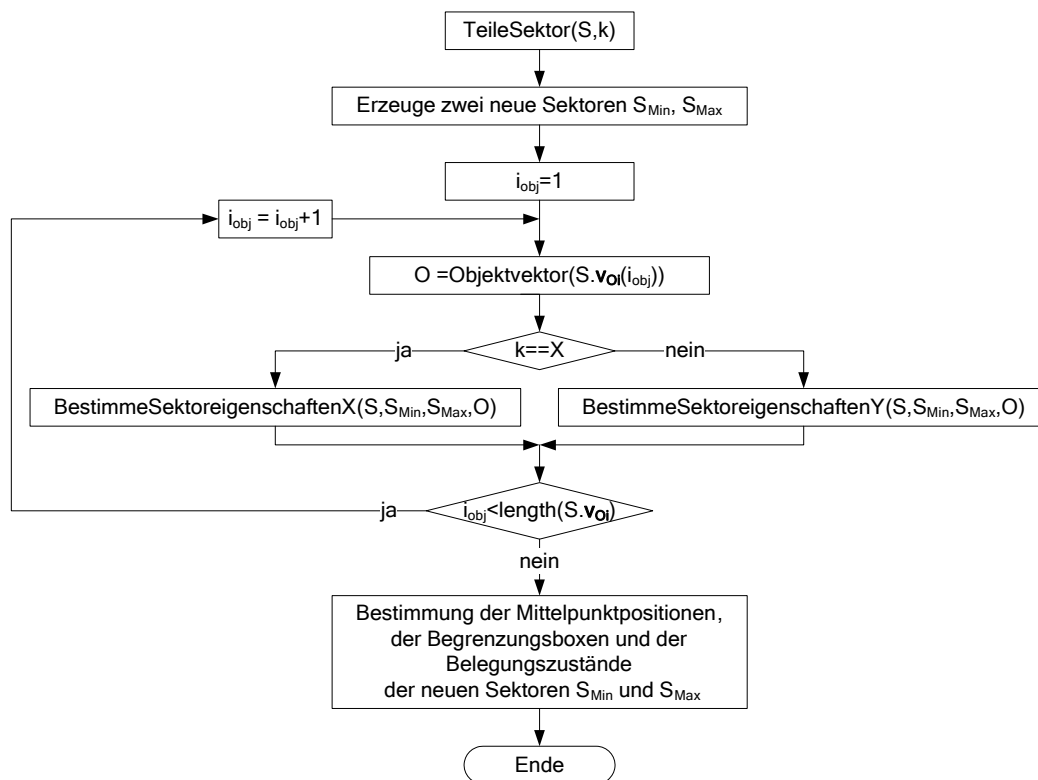


Abbildung 4.10: Algorithmus zur Teilung eines Sektors

#### 4.3.1.2 Ermittlung der Nachbarschaftsbeziehungen

Der vorhergehende Abschnitt stellte die Schritte zur Zerteilung des Operationsgebietes in einzelne Sektoren vor. Die Informationen über diese Zerteilungen sowie die Lage der *freien* Sektoren wurden im dort beschriebenen Zerteilungsalgorithmus als Baum abgespeichert. Der Aufbau dieser Datenstruktur ermöglicht es, die räumlichen Beziehungen der Sektoren sowie ihre „Entstehungsgeschichte“ aus den durchgeführten Zerteilungen abzubilden. Die bei der Bestimmung der Nachbarschaftsbeziehungen durchzuführenden geometrischen Suchoperationen lassen sich mit solchen Bäumen effizient lösen. Ein Baum besteht aus Knoten, die ausgehend von der *Wurzel*, dem so genannten Wurzelknoten, mehrere Nachfolger besitzen können, die ebenfalls Knoten sind. Diese Nachfolger werden auch als *Söhne* bezeichnet. Außer dem Wurzelknoten besitzt jeder Knoten einen Vorgänger, den so genannten *Vater*. Hat ein Knoten keine Nachfolger, wird er als *Blatt* bezeichnet. Besitzen die einzelnen Knoten eines Baumes nur zwei Söhne, heißt der so entstandene Baum auch Binärbaum oder Baum der Ordnung 2 [119].

Der im nachfolgenden Verfahren eingesetzte 2d-Baum gehört zur Gruppe der Binärbäume. Dabei enthalten seine Knoten die Teilungsposition (Splitwert) des zu teilenden Sektors abwechselnd in der x- und der y-Koordinate (Splitkoordinate) [91]. Abbildung 4.12 zeigt einen so entstandenen 2d-Baum für ein vorgegebenes Operationsgebiet mit einem Objekt. Jeder Knoten enthält einen *Schlüsseltyp*, der angibt in welcher Koordinatenrichtung geteilt wird und den eigentlichen *Schlüssel*, den Splitwert. Enthält ein Sektor keine Objekte mehr, finden keine weiteren Teilungen statt und der Knoten wird als Blatt definiert. Dies geschieht in der Funktion *ErzeugeBlatt(S)* innerhalb des Zerteilungsalgorithmus (siehe Abbildung 4.7). Dabei wird der freie Sektor in einen Sektorvektor  $v_s$  aufgenommen und sein Index in den *Sektorindex* des Knoten geschrieben.

Knoten K	
Schlüsseltyp: {NO, X, Y}	
Schlüssel: s	
Sektorindex: $s_{\text{Index}} = -1$	
Zeiger Sohn <sub>Min</sub> =NULL	Zeiger Sohn <sub>Max</sub> =NULL

Abbildung 4.11: Daten eines Knotenelements

Bei der programmtechnischen Realisierung des Baumes müssen auch die Beziehungen zwischen den Knoten hergestellt werden. Dies erfolgt durch einen zeigerbasierten Zugriff auf die einzelnen Knotenelemente. So enthält jeder Knoten für jeden seiner Nachfolger einen Zeiger. Abbildung 4.11 zeigt die einzelnen Daten eines Knotenelements. Zur Bestimmung der Nachbarschaftsbeziehungen ist die Funktion *SucheSektor*( $x, t_{\max}$ ) im Baum implementiert. Die Funktion ermittelt den freien Sektor, in welchem sich die übergebene Position  $x$  befindet. Der zurückgegebene Wert entspricht dabei seinem Sektorindex im Sektorvektor  $v_s$ . Liegt die vorgegebene Position nicht in einem freien Sektor, wird als Sektorindex  $-1$  zurückgegeben. Bei der Suche folgt man durch Größenvergleich der einzelnen Positionskoordinaten von  $x$  mit den entsprechenden Schlüsseln den Knoten bis zum Erreichen eines Blattes oder einer vorgegebenen maximalen Suchtiefe  $t_{\max}$ . Vor der Ermittlung der Nachbarschaftsbeziehungen wird durch die Funktion *SucheSektor*( $x, t_{\max}$ ) für den Start- und Zielpunkt der freie Sektor bestimmt, in dem sie liegen. Dann werden die Kanten zwischen diesen Punkten und dem Mittelpunkt des freien Sektors, sowie den Mittelpunkten der benachbarten, achsenparallel zum Start- bzw. Zielpunkt liegenden, freien Sektoren in den Graphen eingetragen.

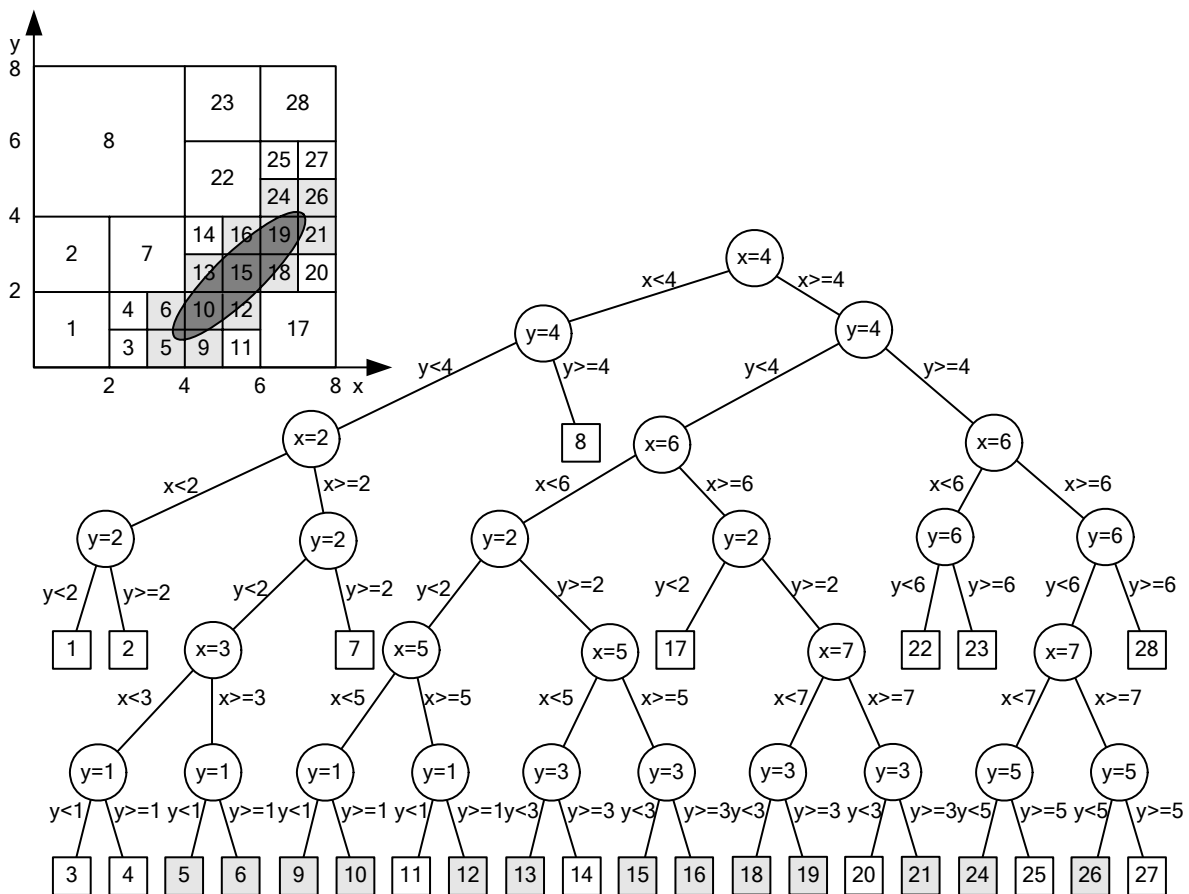


Abbildung 4.12: 2d-Baum des vorgegebenen Operationsgebietes

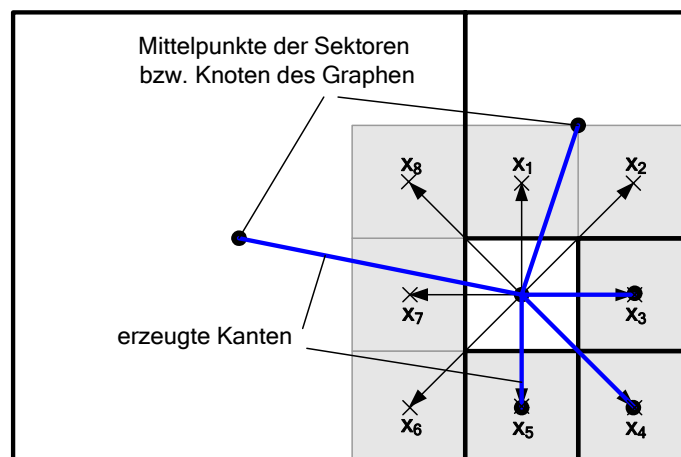


Abbildung 4.13: Bestimmung der Nachbarschaftsbeziehungen

Die Bestimmung der Nachbarschaftsbeziehungen erfolgt in mehreren Stufen, deren Anzahl der maximalen Teilungstiefe eines Sektors entspricht. Dabei werden in einer Stufe die Sektoren von gleicher Größe mit gleichgroßen oder größeren Sektoren auf eine Nachbarschaft überprüft. Begonnen wird in der ersten Stufe mit den kleinsten Sektoren. Bei der Ermittlung der Nachbarschaftsbeziehungen eines Sektors wird untersucht, ob sich die den Sektor umgebenden Gebiete in freien Sektoren befinden. Diese Gebiete sind in Abbildung 4.13 grau gekennzeichnet und haben die gleichen Abmaße wie der untersuchte Sektor. Bei der Untersuchung werden für die einzelnen Mittelpunktpositionen  $x_1$  bis  $x_8$  der Gebiete freie Sektoren im 2d-Baum durch die Funktion  $\text{SucheSektor}(x, t_{\max})$  ermittelt. Die so gefundenen Sektoren sind dann gleichzeitig auch die Nachbarn des überprüften Sektors. Wurde eine Nachbarschaftsbeziehung gefunden, wird die Kante zwischen den Mittelpunkten des überprüften Sektors und des gefundenen Nachbarsektors in den Graphen eingetragen. Durch die gleichen Größenverhältnisse aller Sektoren ist sichergestellt, dass die Kante nur durch freie Sektoren führt. Eine Überprüfung auf Überschneidungen mit den Objekten braucht im Gegensatz zum Sichtbarkeitsgraphen nicht durchgeführt zu werden. Jedes so übergeprüfte Sektorelement wird durch Setzen des Sektorindex  $-1$  im entsprechenden Knoten von der weiteren Überprüfung ausgeschlossen. So wird die weitere Suche vereinfacht und eine Kante zwischen zwei Knoten nicht mehrmals in den Graphen eingetragen. Abbildung 4.14 zeigt die einzelnen Stufen zur Ermittlung der Nachbarschaftsbeziehungen und zur Erzeugung der Kanten für den Graphen.

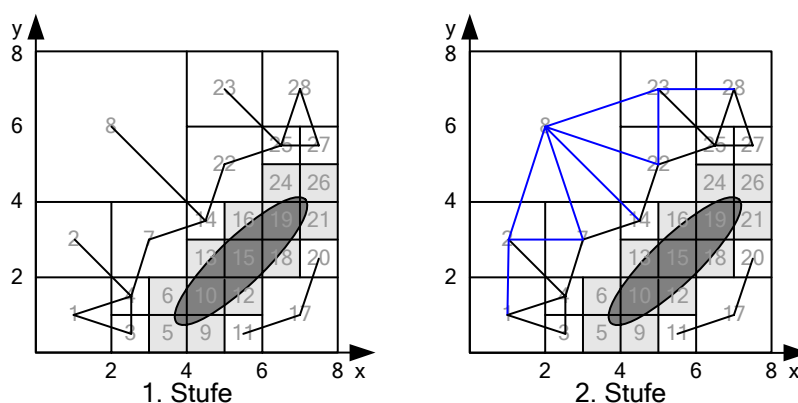


Abbildung 4.14: Einzelne Stufe zur Ermittlung der Nachbarschaftsbeziehungen und zur Erzeugung der Kanten

### 4.3.2 Sichtbarkeitsgraph

Bei einem Sichtbarkeitsgraphen bilden die Ecken der einzelnen Hindernisse die Knoten des Graphen. Gibt es eine Sichtverbindung zwischen zwei Knoten, werden sie durch eine Kante verbunden. So entsteht ein Netz aus befahrbaren Wegabschnitten im Operationsgebiet. Da die verwendete Objektform des elliptischen Zylinders keine Eckpunkte enthält, müssen in einer ersten Phase diese Objekte in eine Polygonform überführt werden, um so die Knoten des Graphen bestimmen zu können. Anschließend erfolgt die Sichtüberprüfung zwischen den einzelnen Knoten. Befinden sich viele Hindernisse im Operationsgebiet, sind sehr viele Sichtüberprüfungen notwendig, um jeden Knoten mit allen anderen Knoten des Graphen zu überprüfen. Für solche Fälle wird das Verfahren des Sektorisierten Sichtbarkeitsgraphen eingesetzt. Dieses Verfahren wurde von Jacobi [86] für die Generierung eines Sichtbarkeitsgraphen unter Echtzeitbedingungen im Rahmen des Projektes „DeepC“ untersucht.

#### 4.3.2.1 Bestimmung der Knoten

Bei der Nachbildung des elliptischen Zylinders durch eine Polygonform muss diese den Zylinder vollständig einschließen. Dabei wird die elliptische Grundfläche durch ein Polygon approximiert, das die Ellipse vollständig überdeckt und möglichst wenig befahrbare Gebiete enthält. Dazu wird das im Abschnitt 5.3.5.3 beschriebene Verfahren zur Bestimmung von Polygonstützstellen um Ellipsen eingesetzt. Bei diesem Verfahren werden die Polygonstützstellen auf Basis normierter Stützstellen, welche um einen Einheitskreis angeordnet sind, berechnet. Die normierten Stützstellen wurden so bestimmt, dass die Kanten des mit diesen Stützstellen gebildeten Polynoms Tangenten des Einheitskreises sind. Durch eine Skalierung und Rotation werden die normierten Stützstellen an die entsprechende Ellipsenform angepasst. Abbildung 4.15 zeigt die so bestimmten Polygonstützstellen um eine Ellipse. In der Vergrößerung ist deutlich das Tangieren der Polygonkanten mit der Ellipse zu erkennen. Wird der Graph für den dreidimensionalen Fall erstellt, werden die ermittelten Polygonstützstellen um den elliptischen Zylinder in vorgegebenen Ebenen angeordnet. Diese befinden sich in der Höhe der Grund- und Deckfläche des elliptischen Zylinders sowie des Start- und des Endpunktes, wenn diese Ebenen sich innerhalb der vertikalen Zylinderausdehnung befinden, und in definierten Abständen entlang dieser Ausdehnung. Bei einer Sektorisierung des Operationsgebietes (siehe Abschnitt 4.3.2.3) stellt die z-Koordinate des Sektormittelpunktes, in dem sich das Hindernis befindet, eine weitere Ebene zur Positionierung der Polygonstützstellen dar [86]. Die so gebildeten Polygonstützstellen sowie der Start- und Zielpunkt werden als Knoten in den Graphen eingetragen.

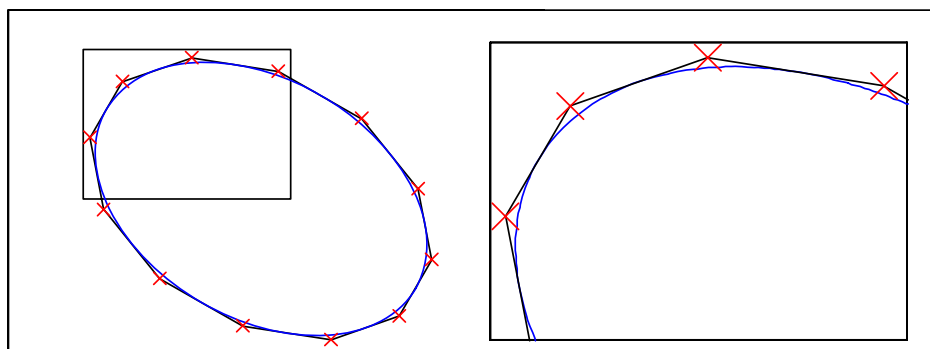


Abbildung 4.15: Polygonstützstellen um eine Ellipse

### 4.3.2.2 Ermittlung der Kanten

Nach dem Einfügen aller Knoten in den Graphen müssen in einer zweiten Phase ihre möglichen Verbindungen untereinander ermittelt und als Kanten in den Graphen eingetragen werden. Eine Verbindung zwischen zwei Knoten ist vorhanden, wenn die Strecke zwischen den beiden Knoten kein Hindernis schneidet. So wird für jede mögliche Verbindung eine Kollisionsüberprüfung mit allen Objekten des Operationsgebietes durchgeführt. Für diese Überprüfung wird das im Abschnitt 2.3.3 dieser Arbeit vorgestellte Zwei-Phasen-Konzept eingesetzt. Dieses Konzept ermöglicht eine schnelle Kollisionsüberprüfung durch die Verwendung einfacher Kollisionstests zur Vorselektion möglicher Kollisionskandidaten. Für einen Graphen  $G=(V, E)$ , der aus  $n_V$  Knoten besteht, sind so:

$$n_E^{Test} = \frac{1}{2} n_V (n_V - 1) \quad (4.2)$$

Kollisionstests durchzuführen. Die Anzahl dieser Tests steigt dabei quadratisch zur Knotenanzahl des Graphen. Dies führt zu einem erheblichen Rechenaufwand, wenn viele Objekte im Operationsgebiet liegen oder der Graph für den dreidimensionalen Raum entworfen wird. Die einzuhaltende Zeitvorgabe zur Generierung des Graphen kann so nicht mehr garantiert werden. Des Weiteren steigt mit der Anzahl der in den Graphen eingetragenen Kanten dessen benötigter Speicherplatz sowie die Rechenzeit bei der Suche nach dem kürzesten Weg. Aus diesem Grund wird das Verfahren des Sektorisierten Sichtbarkeitsgraphen verwendet, welches im nächsten Abschnitt vorgestellt wird.

### 4.3.2.3 Sektorisierung des Operationsgebietes

Um die Anzahl der Sichtüberprüfungen und die Größe des Graphen zu senken, wird beim Verfahren des Sektorisierten Sichtbarkeitsgraphen das Operationsgebiet in einzelne, gleichgroße Sektoren unterteilt [86]. Bei den Sichtüberprüfungen werden nur die Strecken, welche innerhalb eines Sektors oder zweier benachbarter Sektoren liegen, auf Schnitt mit den Objekten in diesen Sektoren überprüft. Diese Vorgehensweise spart erheblichen Rechenaufwand und senkt den Datenumfang des Graphen gegenüber einem ungeteilten Sichtbarkeitsgraphen, was sich jedoch nicht auf die berechneten Routen auswirkt. Durch eine Glättung des gefundenen Weges (siehe Abschnitt 4.4.2) besitzen die ermittelten Routen bei beiden Verfahren einen ähnlichen Verlauf.

Die Zerteilung des Operationsgebietes in Sektoren erfolgt in Analogie zu dem im Abschnitt 4.3.1.1 vorgestellten Verfahren, wobei in diesem Verfahren für jede Koordinatenrichtung eine separate Teilungstiefe verwendet wird. Dies kann dazu führen, dass ein Sektor, der die maximale Teilungstiefe in der y-Koordinatenrichtung durch die vorher durchgeführten Teilungen schon erreicht hat, nur noch in der x-Koordinatenrichtung geteilt werden muss. Durch ein solches Vorgehen sind im Gegensatz zur  $2^n$ -Teilung die Seitenverhältnisse der Sektorgrößen nicht von den Seitenverhältnissen des Operationsgebietes abhängig, wodurch ein beliebig ausgedehntes Operationsgebiet durch Sektoren mit ähnlichen Seitenlängen nachgebildet werden kann. Die maximale Teilungstiefe  $t_{max\_i}$  für die einzelnen  $i$  Koordinaten bestimmt sich unter Verwendung der Seitenlängen des Operationsgebietes  $l_{Area\_i}$  und der geforderten Seitenlängen  $l_{Sektor\_i}$  des Sektors zu:

$$t_{max\_i} = \text{round} \left( \frac{\ln l_{Area\_i} - \ln l_{Sektor\_i}}{\ln 2} \right). \quad (4.3)$$

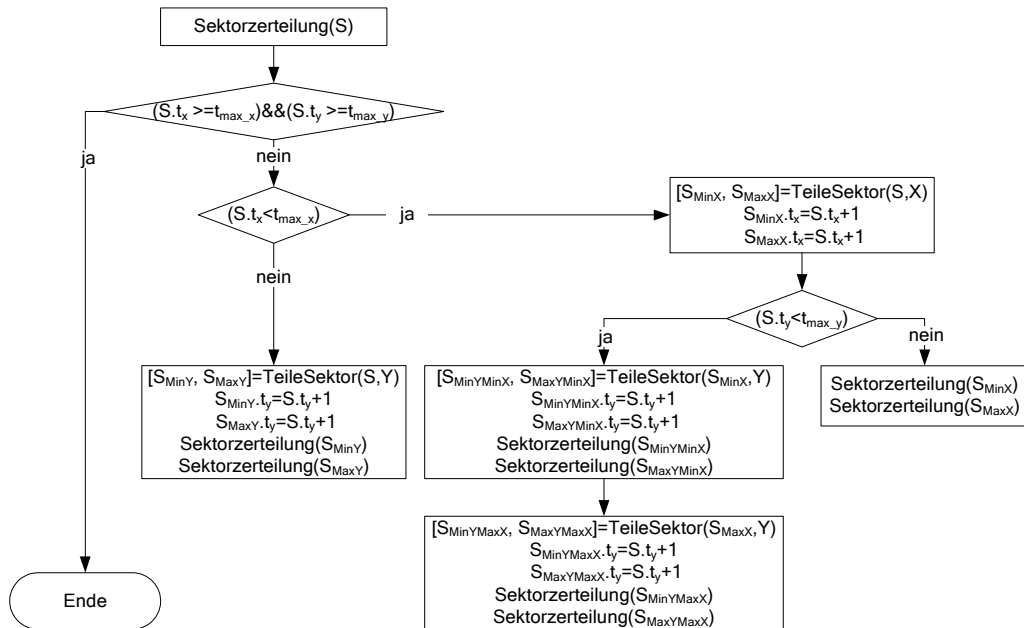


Abbildung 4.16: Flussdiagramm der Sektorzerteilung

Abbildung 4.16 zeigt das Flussdiagramm zur Sektorzerteilung für den zweidimensionalen Raum. Das Flussdiagramm für den dreidimensionalen Raum befindet sich im Anhang. Die Zerteilung wird beendet, wenn beide Teilungstiefen  $t_x$  und  $t_y$  des Sektors die maximalen Teilungstiefen  $t_{max\_x}$  und  $t_{max\_y}$  erreicht haben. Dazu enthält ein Sektorelement, im Unterschied zum Sektorelement beim Quadtreegraphen (vergleiche Abbildung 4.8 und Abbildung 4.17), eine Teilungstiefe für jede Koordinate. Die Funktion und die Bestimmung des Objektindex-Vektors und des Eckenbelegungs-Vektors sind bei beiden Sektorelementen jedoch gleich (siehe Abschnitt 4.3.1.1). Zusätzlich enthält das Sektorelement des Sichtbarkeitsgraphen einen Knotenindex-Vektor  $\mathbf{v}_{Vi}$ , der die Nummern der Knoten des Graphen enthält, die im Sektor liegen. Die Erstellung der einzelnen Sektoren erfolgt vor der Erzeugung der Knoten. So kann jeder in den Graphen eingefügte Knoten zugleich dem entsprechenden Sektor zugeordnet werden. Zusätzlich werden die Mittelpunktpositionen der Sektoren als Knoten in den Graphen aufgenommen. Dadurch wird sichergestellt, dass eine Route auch durch Sektoren, die keine Objekte und damit auch keine Knoten enthalten, geführt wird. Zur Bestimmung der Kanten des Graphen werden zuerst die Verbindungen der Knoten innerhalb der einzelnen Sektoren auf Kollision mit den darin befindlichen Objekten überprüft. Anschließend wird die Kollisionsüberprüfung zwischen den Knoten benachbarter Sektoren durchgeführt. Erfolgt die Überprüfung zwischen achsenparallel zueinander liegenden Sektoren, sind die auf Kollision zu testenden Objekte die Objekte in beiden Sektoren. Liegen hingegen die Sektoren diagonal zueinander, müssen zusätzlich noch die Objekte der an beide Sektoren angrenzenden Sektoren bei der Überprüfung einbezogen werden.

Sektorelement S
Mittelpunktposition: $\mathbf{x}$
Teilungstiefe: $t_x, t_y, t_z$
Achsengerichtete Begrenzungsbox: $\mathbf{x}_{min} \mathbf{x}_{max}$
Objektindex-Vektor: $\mathbf{v}_{Oi}$
Eckenbelegungs-Vektor: $\mathbf{v}_{Eb}$
Knotenindex-Vektor: $\mathbf{v}_{Vi}$

Abbildung 4.17: Eigenschaften des Sektorelements für den Sichtbarkeitsgraphen

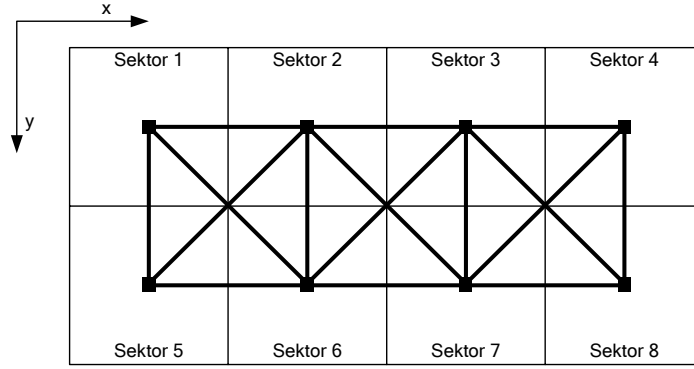


Abbildung 4.18: Schema zur Überprüfung benachbarter Sektoren

So kann z.B. bei dem in Abbildung 4.18 dargestellten sektorisierten Operationsgebiet sichergestellt werden, dass eine Verbindung zwischen einem im Sektor 1 und Sektor 6 gelegenen Knoten, welche auch über die Sektoren 2 oder 5 verläuft, mit deren Objekten auf Kollision überprüft wird. Die zu überprüfenden Kombinationen zwischen benachbarten Sektoren sind in Abbildung 4.18 für  $n_{\text{Sektor}_x} = 4$  und  $n_{\text{Sektor}_y} = 2$  dargestellt. In Annahme, dass alle  $n_v$  Knoten gleich verteilt im Operationsgebiet liegen, ergibt sich die Anzahl durchzuführender Tests beim sektorisierten Sichtbarkeitsgraphen im zweidimensionalen Raum aus der Anzahl der Sektoren:

$$n_{\text{Sektoren}} = n_{\text{Sektor}_x} n_{\text{Sektor}_y} \quad (4.4)$$

und aus der Anzahl der möglichen Kombinationen zwischen benachbarten Sektoren:

$$n_{\text{Sektorenpaare}} = \underbrace{n_{\text{Sektor}_x} (n_{\text{Sektor}_y} - 1)}_{\text{waagrecht liegende Sektorenpaare}} + \underbrace{(n_{\text{Sektor}_x} - 1) n_{\text{Sektor}_y}}_{\text{senkrecht liegende Sektorenpaare}} + \underbrace{2(n_{\text{Sektor}_x} - 1)(n_{\text{Sektor}_y} - 1)}_{\text{diagonal liegende Sektorenpaare}} \quad (4.5)$$

zu:

$$n_{E\_sektorisiert}^{\text{Test}} = \underbrace{n_{\text{Sektoren}} \left( \frac{1}{2} \frac{n_v}{n_{\text{Sektoren}}} \left( \frac{n_v}{n_{\text{Sektoren}}} - 1 \right) \right)}_{\text{Anzahl der Tests in den einzelnen Sektoren}} + \underbrace{n_{\text{Sektorenpaare}} \left( \frac{n_v}{n_{\text{Sektoren}}} \right)^2}_{\text{Anzahl der Tests zwischen benachbarten Sektoren}} \quad (4.6)$$

Abbildung 4.19 zeigt das Verhältnis der durchzuführenden Tests zwischen dem sektorisierten Sichtbarkeitsgraphen zum ungeteilten Sichtbarkeitsgraphen. Hier ist eine deutliche Abnahme der Tests mit zunehmender Sektorzerteilung zu erkennen.

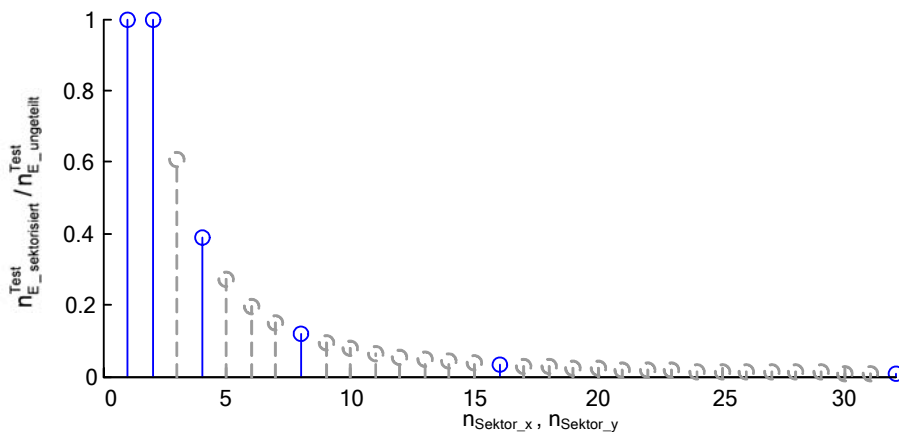


Abbildung 4.19: Vergleich der durchzuführenden Tests beim sektorisierten und ungeteilten Sichtbarkeitsgraphen

### 4.3.3 Bestimmung des Kostenwertes

Bei der Erstellung des Graphen muss für jede ermittelte Kante ihr Kostenwert bestimmt werden. Im einfachsten Fall wird dabei die euklidische Distanz zwischen den beiden Knoten der Kante bestimmt. Um die Strömungsinformation mit in die Bewertung der Kante einfließen zu lassen, wird als Kriterium die benötigte Zeit  $t_{Bahn}$  zum Abfahren der Kante verwendet. Diese Zeit bestimmt sich für die  $i$ -te Kante durch Bildung des Quotienten aus der Weglänge  $s_{Bahn}$  und der Geschwindigkeit  $v_{Bahn\_ef}$ , mit der das Fahrzeug auf diesem Wegabschnitt gegenüber einem erdfesten Koordinatensystem fährt. Es gilt:

$$t_{Bahn}^i = \frac{s_{Bahn}^i}{v_{Bahn\_ef}^i}. \quad (4.7)$$

Diese Geschwindigkeit  $v_{Bahn\_ef}$  ist abhängig von der Fahrzeuggeschwindigkeit durchs Wasser  $v_{Fahr\_kf}$  (Marschgeschwindigkeit), dem Betrag und der Richtung des Strömungsvektors  $\mathbf{v}_{Strömung}$  sowie der Richtung des Wegabschnittes  $\mathbf{v}_{Bahn}$ . Sie kann durch eine geometrische Schnittberechnung zwischen einer Linie und einem Kreis (2D) bzw. Kugel (3D) [135] auf Basis von Abbildung 4.20 nach den folgenden Beziehungen ermittelt werden:

$$\text{Linie: } \mathbf{x}(v_{Bahn\_ef}) = v_{Bahn\_ef} \mathbf{v}_{Bahn}^0 \quad (4.8)$$

$$\text{Kreis / Kugel: } v_{Fahr\_kf}^2 = \|\mathbf{x} - \mathbf{v}_{Strömung}\|^2$$

$$disc = \left( \mathbf{v}_{Bahn}^0{}^T \cdot \mathbf{v}_{Strömung} \right)^2 + v_{Fahr\_kf}^2 - \mathbf{v}_{Strömung}{}^T \cdot \mathbf{v}_{Strömung}$$

für  $disc > 0$ : (4.9)

$$v_{Bahn\_ef} = \mathbf{v}_{Bahn}^0{}^T \cdot \mathbf{v}_{Strömung} + \sqrt{disc}.$$

Wird die Diskriminante negativ bedeutet dies, dass das Fahrzeug nicht mehr auf dem Wegabschnitt gehalten werden kann (siehe Abbildung 4.21). Ist die Geschwindigkeit  $v_{Bahn\_ef}$  negativ, hält sich das Fahrzeug noch auf dem Wegabschnitt, fährt aber rückwärts (siehe Abbildung 4.22). Beide Fälle müssen durch Setzen eines großen Zahlenwertes für das Kantengewicht berücksichtigt werden. So werden solche Wegabschnitte bei der Wegesuche ausgeschlossen, und es kommt nicht dazu, dass ein Fahrzeug in eine starke Gegenströmung gerät und dadurch die Sollroute verlässt.

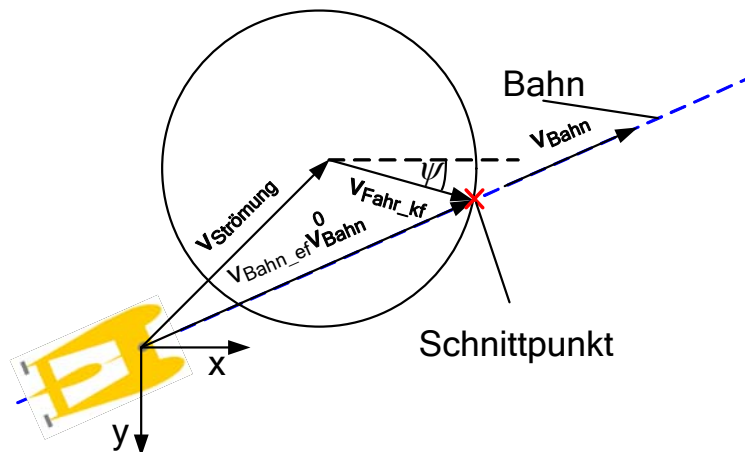


Abbildung 4.20: Definition der Geschwindigkeitsvektoren



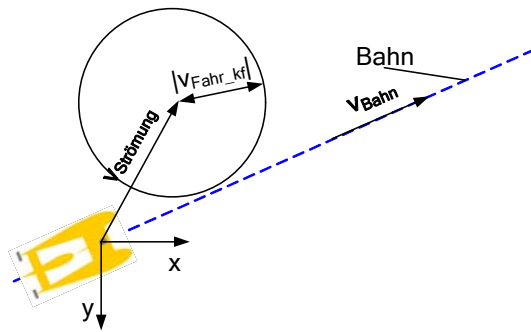
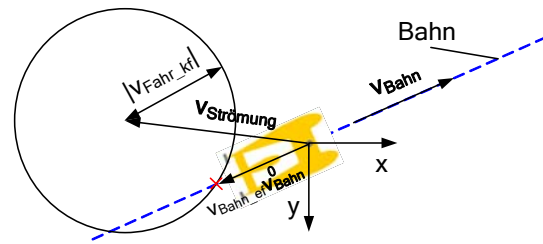


Abbildung 4.21: Negative Diskriminante

Abbildung 4.22:  $v_{Bahn\_ef}$  ist negativ

Bei Verwendung der oben beschriebenen Kostenfunktion ist es auch möglich, bei komplexen Strömungsprofilen einen zeitoptimalen Weg zu ermitteln. Dies soll an dem nachfolgenden Beispiel erläutert werden.

Die Aufgabe besteht in der Ermittlung eines zeitoptimalen Weges bei der Durchquerung eines Flusslaufes, welcher das nachfolgende Strömungsprofil besitzt:

$$\begin{aligned}
 v_{Strömung}^x &= 0.0 \\
 v_{Strömung}^y &= \frac{4}{b^2} x(b-x) v_{Strömung}^0 \\
 \text{mit } b &= 300\text{m} \quad v_{Strömung}^0 = 1.8 \frac{\text{m}}{\text{s}} .
 \end{aligned} \tag{4.10}$$

Abbildung 4.23 zeigt die ermittelten Wegverläufe bei Verwendung der Graphenmethode sowie die korrekte zeitoptimale Lösung durch die Optimale Steuerung [30, 120]. Beide Wegverläufe besitzen ähnliche Profile, die durch jeden Schwimmer, der einen Fluss mit starker Strömung durchschwommen hat, bestätigt werden. Zuerst schwimmt man gegen die Strömung, um sich dann in der Mitte des Flusses treiben zu lassen und in Richtung des anderen Ufers zu schwimmen. Der Graph wurde bei dieser Aufgabe durch ein Gitternetz aus Knoten aufgebaut. Eine feinere Diskretisierung dieses Netzes würde zu einer weiteren Annäherung an die exakte Lösung führen.

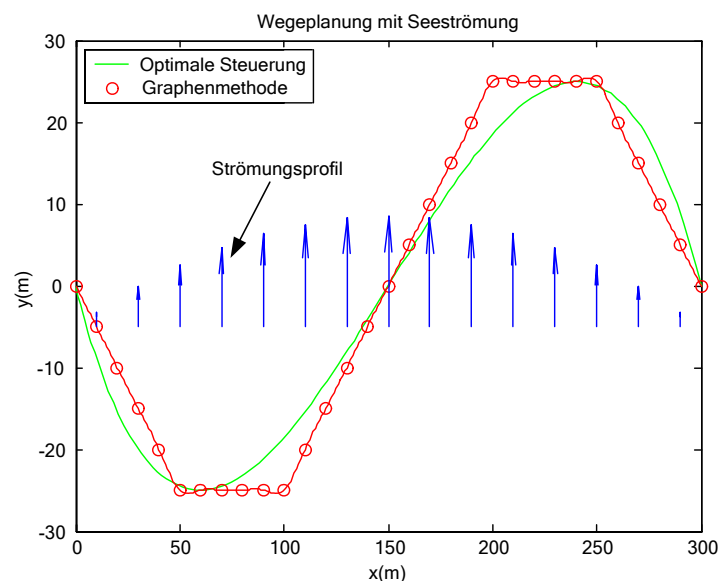


Abbildung 4.23: Wegeplanung mit Strömung

## 4.4 Erzeugung des Routenplanes

### 4.4.1 Routenplanung mit der Boost Graph Library

Zur Speicherung des Graphen und zur Ermittlung des kürzesten Weges wurde die Boost Graph Library (BGL) [20, 138] Version 1.33.1 eingesetzt. So konnte auf vorhandene und frei verfügbare Software zurückgegriffen werden, die von einer Vielzahl von C++ Compilern unterstützt wird. Durch die Verwendung der Techniken der Generischen Programmierung [3] und der Standard Template Library (STL) [99] in der BGL steht mit ihr eine leistungsstarke, flexible und einfach erweiterbare Software zur Verfügung.

#### 4.4.1.1 Datenstrukturen

Die BGL stellt zur Haltung des Graphen die *Adjazenzmatrix* und die *Adjazenzliste* zur Verfügung [119]. Der Vorteil der *Adjazenzmatrix* besteht im Einfügen und Entfernen von Knoten in konstanter Zeit  $O(1)$ . Dem stehen einige Nachteile gegenüber. Der erste Nachteil besteht im erhöhten Speicherbedarf von  $O(|V|^2)$  anstelle von  $O(|V|+|E|)$  bei der *Adjazenzliste* für einen Graphen  $G=(V, E)$  ( $|V|$ : Anzahl der Knoten,  $|E|$ : Anzahl der Kanten). Ein weiterer Nachteil besteht im Aufsuchen aller benachbarten Knoten bei einer geringen Kantenanzahl, so wie sie bei der Breitensuche durchgeführt wird [138]. Nach [135] sollte eine *Adjazenzmatrix* für dicht besetzte Graphen eingesetzt werden, wo  $|E| \approx |V|^2$  ist. Bei dünn besetzten Graphen, wo  $|E|$  viel kleiner als  $|V|^2$  ist, sollten Adjazenzlisten verwendet werden. Dieses Verhältnis liegt bei den in Abschnitt 4.3 erzeugten Graphen vor, so dass in dieser Arbeit die *Adjazenzliste* zur Speicherung des Graphen eingesetzt wird. Abbildung 4.24 zeigt die Darstellung eines ungerichteten Graphen in einer *Adjazenzliste* und einer *Adjazenzmatrix*.

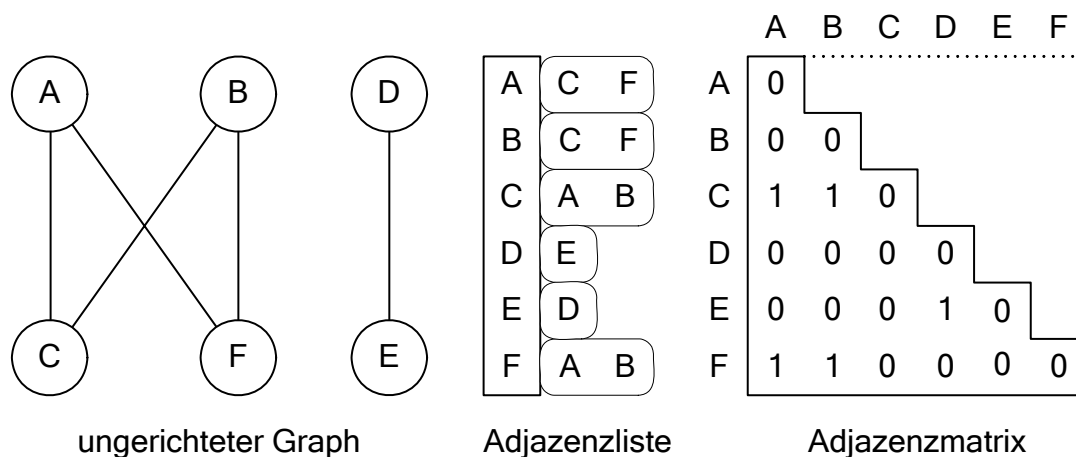


Abbildung 4.24: Speicherung eines ungerichteten Graphen in einer Adjazenzliste und Adjazenzmatrix [138]

#### 4.4.1.2 Suchverfahren

Die BGL bietet eine Vielzahl von Suchalgorithmen in Graphen an. Zur Ermittlung des kürzesten Weges werden in dieser Arbeit der Dijkstra-Algorithmus und der A\*-Algorithmus eingesetzt. Zur detaillierten Beschreibung der Algorithmen wird auf [16, 41, 117, 138] verwiesen.

## 4.4.2 Verfahren zur Glättung des Weges

Die mit den in dieser Arbeit verwendeten Verfahren zur Graphenerzeugung gebildeten geometrischen Graphen sind nicht vollständig. Das heißt, dass nicht jeder Knoten mit jedem anderen Knoten innerhalb des Graphen über eine Kante verbunden ist, auch wenn diese durch die Kollisionsfreiheit zwischen der Verbindung und den Hindernissen möglich wäre. So können die Suchverfahren auch nur die Kanten bei ihrer Suche berücksichtigen, die im Graphen enthalten sind. Eine direkte Verbindung, die zwischen zwei Wegpunkten innerhalb des ermittelten Routenplanes möglich wäre, gibt es deshalb nicht. Dies wird vor allem beim Quadtree-Graphen deutlich, bei denen es nur Kanten zwischen benachbarten Sektoren gibt. Ein so abzufahrender Routenplan ist dadurch gekennzeichnet, dass es viele Streckenmanöver mit Richtungsänderungen gibt, denen bei kurzen Streckenlängen und den damit verbundenen engen Wenderadien nicht zu folgen ist. Außerdem besitzen solche stufen- bzw. schlangenlinienförmige Routen keinen wegoptimalen Verlauf. Deshalb wird nachfolgend ein Verfahren vorgestellt, welches den Routenplan glätten soll, ohne dass es dabei zu Überschneidungen mit Objekten kommt. In [101] wurden zwei Methoden für die Glättung eines mit einem Quadtree-Graphen erzeugten Routenplanes auf Basis heuristisch aufgestellter Regeln und numerischer Optimierung entwickelt. Dabei können einzelne Wegpunkte gelöscht oder innerhalb ihres Sektors verschoben werden. Eine Überschneidungsüberprüfung der Route mit belegten Sektoren findet bei diesen Methoden nicht statt.

Die Abbildung 4.25 zeigt das Flussdiagramm für das entwickelte Verfahren zur Routenglättung. Der bei der Wegesuche ermittelte Routenplan wird durch eine Liste von Wegpunkten  $\mathbf{v}_{\text{Wegpunkte\_org}}$  beschrieben. Bei dem Verfahren wird der Startpunkt  $\mathbf{x}_{\text{Start}}$  dieser Liste auf eine direkte Verbindung mit den nachfolgenden Wegpunkten überprüft. Diese Überprüfung beginnt beim dritten Wegpunkt des Routenplanes und wird solange fortgeführt, bis es eine Kollision zwischen den Objekten und dem aktuell überprüften Streckenelement gibt. Diese Strecke beschreibt die direkte Verbindung zwischen dem Startpunkt  $\mathbf{x}_{\text{Start}}$  und dem aktuell zu überprüfenden Wegpunkt  $\mathbf{v}_{\text{Wegpunkte\_org}}[i]$ . Liegt eine Kollisionssituation vor, wird der vorhergehende Wegpunkt  $\mathbf{v}_{\text{Wegpunkte\_org}}[i-1]$  in den neuen Wegpunktvektor  $\mathbf{v}_{\text{Wegpunkte}}$  übernommen und als neuer Startpunkt  $\mathbf{x}_{\text{Start}}$  definiert. Mit ihm wird die Überprüfung an dieser Stelle des Routenplanes wie oben beschrieben fortgesetzt. So wird der gesamte Routenplan bis zum Schluss überprüft. Als Ergebnis erhält man einen neuen Wegpunktvektor  $\mathbf{v}_{\text{Wegpunkte}}$ , der sich durch weniger Wegpunkte als der zu überprüfende Wegpunktvektor  $\mathbf{v}_{\text{Wegpunkte\_org}}$  auszeichnet. Dieser oben beschriebene Ablauf wird mit dem neu gebildeten Wegpunktvektor solange wiederholt, bis sich keine Streckenabschnitte mehr zusammenfassen lassen bzw. die Wegpunktanzahl des Routenplanes bei einem Durchlauf nicht mehr gesenkt wird. Eine solche wiederholte Überprüfung ist notwendig, um eine vorhandene „Nische“, wie sie in Abbildung 4.26 dargestellt ist, zu glätten. Eine solche Nische entsteht, wenn in Durchlaufrichtung des Routenplanes eine direkte Sicht auf den übernächsten Wegpunkt durch ein Objekt verdeckt wird. Deshalb findet immer ein wechselseitiges Vorwärts- und Rückwärtsdurchlaufen des Routenplanes mit dem oben beschriebenen Algorithmus statt. So können diese Nischen mit weniger Durchläufen als nur beim einseitigen Vorwärtsdurchlaufen entfernt werden. Soll der generierte Routenplan zeitoptimal sein und die Strömungsinformation berücksichtigen, muss nach jeder erfolgreichen Kollisionsüberprüfung eine Überprüfung der Zeiten zum Abfahren des neu gebildeten Streckenabschnittes zu den Zeiten der beiden zusammengefassten Teilstrecken durchgeführt werden.

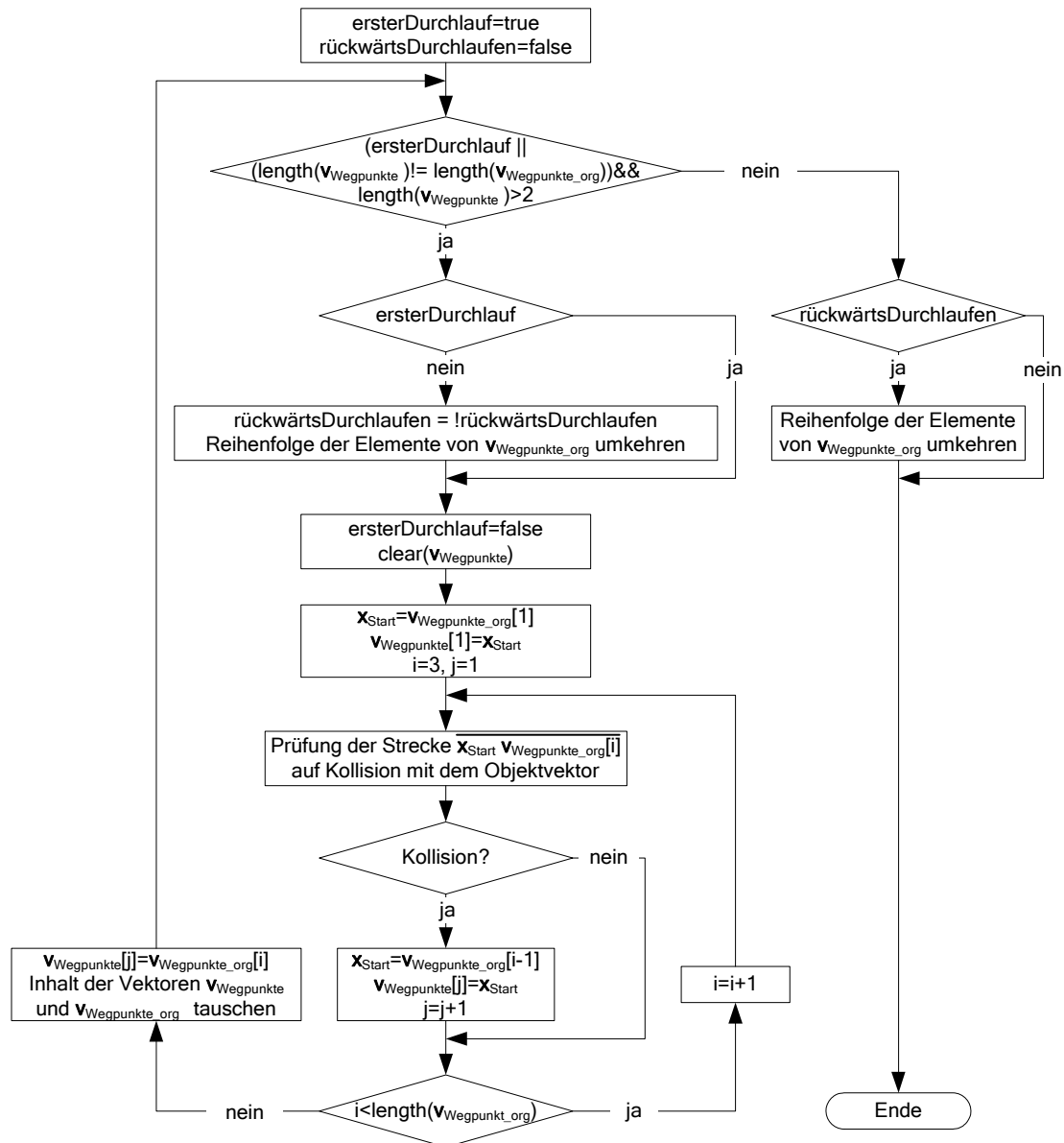


Abbildung 4.25: Flussdiagramm zur Glättung des Routenplanes

Nur wenn es eine Senkung der Abfahrzeit gibt, ist eine Zusammenfassung der Teilstrecken möglich. Die Kollisionserkennung der Streckenabschnitte mit den elliptischen Objekten erfolgt unter Verwendung des in Kapitel 2 vorgestellten Zwei-Phasen-Konzeptes. Dies ermöglicht eine schnelle Ermittlung der Kollisionssituationen durch eine entsprechende Vorselektion von möglichen Kollisionskandidaten.

Abbildung 4.26 zeigt die einzelnen Durchläufe des Glättungsalgorithmus in einem Operationsgebiet mit zwei Hindernissen. Durch Verwendung des Glättungsalgorithmus konnte die Anzahl der Streckenabschnitte von 17 auf 2 gesenkt werden. Die Gesamtstrecke des neuen Routenplanes wurde gegenüber der vom Suchverfahren ermittelten Route um 3 % gesenkt. Das Zusammenfassen der Streckenabschnitte führt zu geringeren Richtungswechseln zwischen den Streckenmanövern. Ein solcher Routenplan kann von den Versuchsträgern mit ihrer begrenzten Manövrierfähigkeit leichter abgefahren werden als ein ungeglätteter Routenplan.

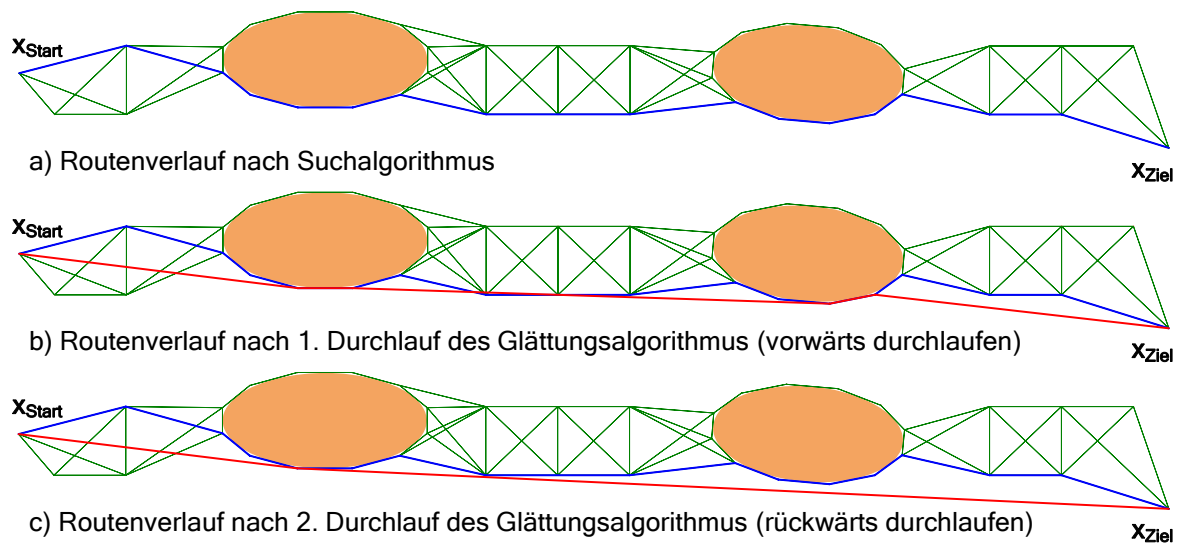


Abbildung 4.26: Schritte zur Glättung eines Routenplanes

## 4.5 Erzeugung des Manöverplanes

### 4.5.1 Erzeugung des Manöverplanes

Der vom Suchalgorithmus erzeugte Routenplan bzw. seine geglättete Form besteht aus einzelnen Wegpunkten, welche die abzufahrenden Streckenabschnitte beschreiben. Findet ein Richtungswechsel zwischen zwei Streckenabschnitten statt, kann ein Fahrzeug den neuen Vorgaben nur verzögert folgen. Um einen stetigen Verlauf beim Abfahren der einzelnen Wegpunkte zu erreichen, wird vielfach die Splineinterpolation eingesetzt [58, 101, 123]. Dabei werden aus den einzelnen Wegpunkten Splinekurven generiert, welche die abzufahrenden Bahnen darstellen. In dieser Arbeit wird ein anderer Ansatz zum Erreichen eines kontinuierlichen Bahnverlaufes verwendet. Hierbei werden zwischen den einzelnen Streckenabschnitten Kreisbogenelemente eingefügt, die das Bewegungsverhalten des Fahrzeuges zum Einfahren auf das neue Streckenelement beschreiben sollen. So wird ein Manöverplan mit den im Abschnitt 1.3.3 beschriebenen Basismanövern Strecke und Kreisbögen aufgebaut. Die Gründe für die Wahl dieses Verfahrens gegenüber dem Spline-Ansatz sind:

- Die Manöverelemente Strecke und Kreisbogen können leicht auf Kollision mit den Objekten überprüft werden. Dies geschieht nach der Erstellung des Manöverplanes und während seiner Abarbeitung bei Erhalt neuer Objektdaten.
- Die Bestimmung der Bahnablage für die Manöver Strecke und Kreisbogen ist gegenüber einer Splinekurve einfacher.
- Eine externe Überprüfung der generierten Manöverliste durch eine *Digitale Seekarte* ist durch die verwendeten Basismanövertypen leicht möglich.
- Die Repräsentation der abzufahrenden Route durch Strecken und Kreisbögen ist eine in der Praxis übliche Form [58].
- Die Berechnung der Kreisbogenelemente ist einfach und die generierten Trajektorienverläufe entsprechen zum größten Teil den vorgegebenen Streckenelementen. Durch die wegoptimalen Vorgaben bei den Suchverfahren werden die Streckenelemente nah an den Objekten vorbeigeführt, so dass es bei der Verwendung eines Spline-Ansatzes zu Überschneidung zwischen den generierten Splinekurven und den Objekten kommen kann.

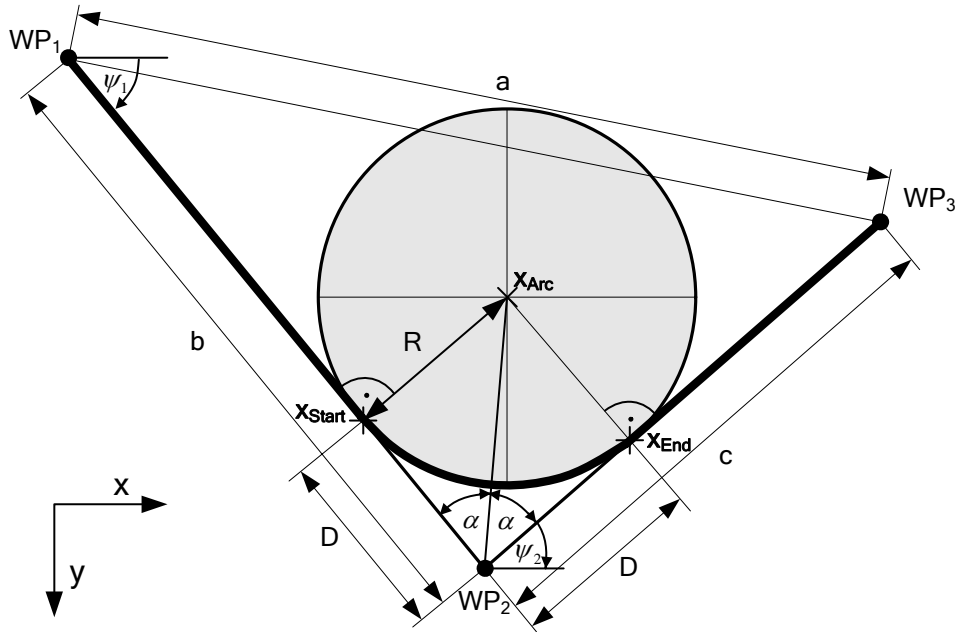


Abbildung 4.27: Geometrische Beziehungen zur Bestimmung des Kreisbogenelementes [58]

Abbildung 4.27 zeigt zwei Streckenabschnitte, die durch die Wegpunktpositionen  $WP_1$ ,  $WP_2$  und  $WP_3$  beschrieben werden. Zwischen diesen beiden Streckenabschnitten soll ein Kreisbogenelement so platziert werden, dass ein stetiger Richtungswechsel beim Einfahren in ein neues Streckenelement entsteht. Die Bestimmung des Kreisbogenmanövers erfolgt für die horizontale Ebene nach den Ausführungen in [58]. Ein solches Kreisbogenmanöver wird durch eine Mittelpunkt-, Start- und Endposition  $x_{Arc}$ ,  $x_{Start}$  und  $x_{End}$  sowie eine Drehrichtung definiert. Der Radius  $R$  des Kreisbogenmanövers ist der minimale Wenderadius des Fahrzeuges. Die Länge  $D$  in Abbildung 4.27 definiert den Beginn bzw. das Ende des Einfahrvorganges. Sie kann durch die Winkelbeziehung:

$$\tan \alpha = \frac{R}{D} \quad D = \frac{R}{\tan \alpha} \quad (4.11)$$

ermittelt werden. Die Bestimmung des Winkels  $\alpha$  erfolgt unter Verwendung des Kosinussatzes für das von den Wegpunkten aufgespannte Dreieck mit den Seitenlängen  $a$ ,  $b$  und  $c$ :

$$\begin{aligned} a^2 &= b^2 + c^2 - 2bc \cos 2\alpha \\ \alpha &= \frac{1}{2} \arccos \left( \frac{b^2 + c^2 - a^2}{2bc} \right). \end{aligned} \quad (4.12)$$

Nach der Berechnung der Länge  $D$  muss überprüft werden, ob sie kleiner als die halbe Weglänge des kürzeren Streckenabschnittes ist. Dies ist notwendig, damit das Kreisbogenmanöver auf dem Streckenabschnitt auch begonnen bzw. beendet werden kann. Ist die halbe Weglänge des kürzeren Streckenabschnittes kleiner als die Länge  $D$ , wird ihr Wert für  $D$  verwendet. Dies führt dazu, dass der Radius  $R$  des Kreisbogenmanövers nach Gleichung (4.11) auch kleiner gewählt werden muss. Ein solcher Fall tritt bei der Umfahrung eines Objektes auf, wo das Umfahrmanöver durch kurze Streckenabschnitte nachgebildet wird. Sonst ist eine solche Senkung der Radiuslänge aufgrund der durchgeführten Glättung des Routenplanes und der damit entstehenden großen Streckenabschnitte sehr selten. Erst wenn Richtungswechsel von mehr als  $90^\circ$  durchgeführt werden, überschreitet  $D$  die Länge des

gewählten Radius  $R$ . Geht man davon aus, dass durch die Glättung solche großen Richtungswechsel nicht vorkommen und die Strecken nicht kleiner als  $2R$  sind, ist eine Verkleinerung des Radius  $R$  nicht notwendig. Die Start- und Endposition des Kreisbogenmanövers, die gleichzeitig die Endposition des ersten Streckenmanövers bzw. die Startposition des zweiten Streckenmanövers ist, wird nach den folgenden Beziehungen bestimmt:

$$\begin{aligned} x_{start} &= x_{WP2} - D \cos \psi_1 \\ y_{start} &= y_{WP2} - D \sin \psi_1 \\ x_{End} &= x_{WP2} + D \cos \psi_2 \\ y_{End} &= y_{WP2} + D \sin \psi_2 \end{aligned} \quad (4.13)$$

Die Festlegung der Drehrichtung und die Bestimmung der Mittelpunktposition  $x_{Arc}$  erfolgt durch Auswertung des Winkels zwischen dem Normalenvektor des ersten Streckenabschnittes  $[-\sin \psi_1 \cos \psi_1]$  und dem Richtungsvektor des zweiten Streckenabschnittes  $[\cos \psi_2 \sin \psi_2]$  nach den folgenden Beziehungen:

$$\left. \begin{aligned} &Drehrichtung = positiv \\ &x_{Arc} = x_{End} - \sin \psi_2 \\ &y_{Arc} = y_{End} + \cos \psi_2 \end{aligned} \right\} \text{für } (0 < (\cos \psi_1 \sin \psi_2 - \sin \psi_1 \cos \psi_2)) \quad (4.14)$$

$$\left. \begin{aligned} &Drehrichtung = negativ \\ &x_{Arc} = x_{End} + \sin \psi_2 \\ &y_{Arc} = y_{End} - \cos \psi_2 \end{aligned} \right\} \text{für } (0 \geq (\cos \psi_1 \sin \psi_2 - \sin \psi_1 \cos \psi_2))$$

Mit diesen oben dargelegten Gleichungen wird der gesamte Routenplan abgearbeitet und ein Manöverplan aus einzelnen Strecken- und Kreisbogenmanövern aufgebaut. Abbildung 4.28 zeigt einen so erstellten Manöverplan. Wird der Manöverplan für Tiefenvorgaben mit absoluter Tiefe erstellt (siehe Abschnitt 3.2.4), entspricht die z-Position des Kreisbogenmanövers der des jeweils mittleren Wegpunktes  $WP_2$  zwischen den zwei Streckenabschnitten.

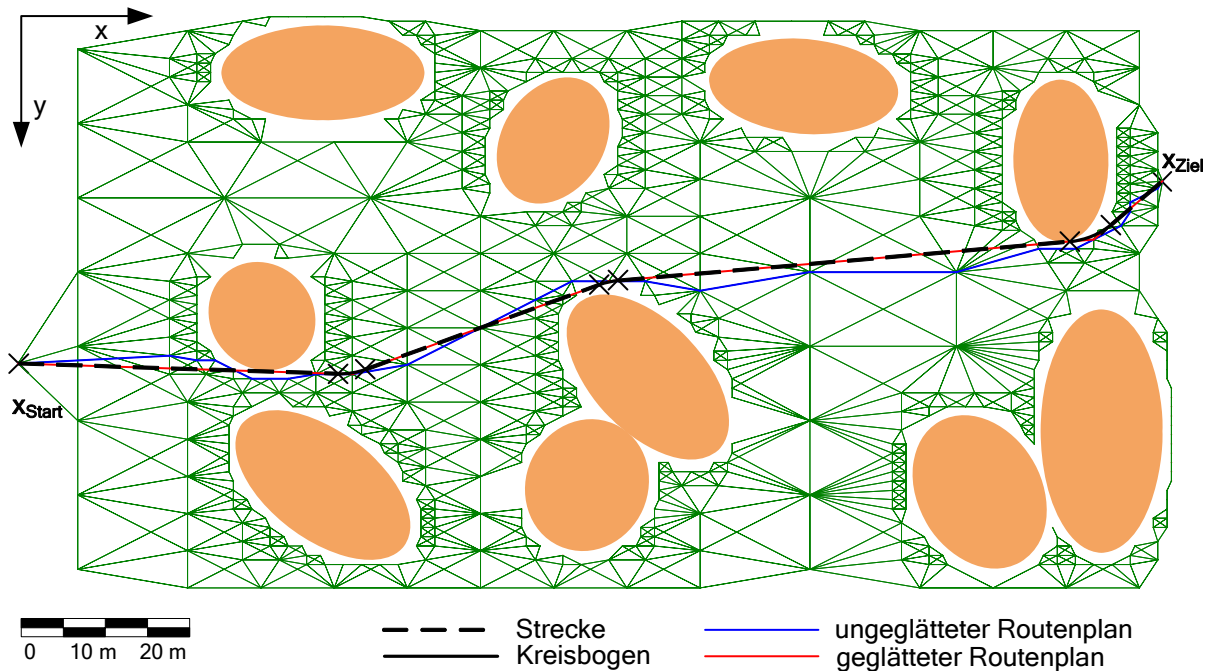


Abbildung 4.28: Erzeugter Manöverplan

## 4.5.2 Überprüfung des Manöverplanes

Die Überprüfung des Manöverplanes erfolgt in einem zweistufigen Prozess. In der ersten Stufe wird der erstellte Plan auf eine Kollision mit der aktuellen Objektliste überprüft. Diese Überprüfung ist notwendig, da der Manöverplan aus dem Routenplan durch Einfügen von Kreisbogenelementen zwischen den einzelnen Streckenübergängen aufgebaut wurde. So weist der Manöverplan an diesen Übergängen einen anderen Verlauf anstelle des ursprünglich vom Routenplan vorgegebenen Streckenverlaufes auf. Durch die wegoptimale Forderung bei der Routenplanung werden die Streckenabschnitte nah an den Objekten vorbeigeführt, wobei es zur Überschneidung zwischen diesen eingefügten Kreisbogenelementen und den elliptischen Zylindern der Sicherheitsbereiche kommen kann. Eine solche Überschneidung, die zu einer Verwerfung des Manöverplanes führen würde, tritt aus den nachfolgenden Gründen jedoch sehr selten auf:

- Durch die Berücksichtigung des minimalen Wenderadius bei der Definition des Sicherheitsabstandes (siehe Abschnitt 1.3.2) werden die Haupt- und Nebenachsen der elliptischen Grundflächen der Sicherheitsbereiche so groß, dass ihre Krümmungsradien größer als die Radien der Kreisbogenmanöver sind.
- Das Umfahren eines Objektes wird durch kurze Streckenabschnitte nachgebildet, was nach Abschnitt 4.5.1 zu einer Verringerung des Kreisbogenradius führt.

Bei den in dieser Arbeit untersuchten Testszenarien kam es durch den vorgegebenen Sicherheitsabstand sowie den minimalen Wenderadius und die gewählten Objektmaße nicht zu Überschneidungen. Weitere Maßnahmen zur Vermeidung solcher Überschneidungen sind:

- ein gegenüber dem Sicherheitsbereich (mit diesem Bereich findet die Kollisionsüberprüfung statt) größer gewählter Objektbereich bei der Erstellung des Graphen,
- ein kleiner gewählter Wenderadius zur Verringerung des Kreisbogenradius bei der Erstellung des Manöverplanes oder
- eine Verringerung des Kreisbogenradius bei einer festgestellten Überschneidung zwischen einem Kreisbogenmanöver und einem Objekt bei der Manöverplanüberprüfung.

Bei einer Verringerung des Kreisbogenradius unterhalb des minimalen Wenderadius kann das Fahrzeug jedoch nicht exakt der vorgegebenen Bahn beim Einfahren auf ein neues Streckenelement folgen. Dies äußert sich darin, dass das Fahrzeug erst verzögert auf das neue Streckenelement einfährt. Eine Kollisionssituation mit dem zu umfahrenden Objekt besteht jedoch nicht, da sich das Fahrzeug durch seinen Wenderadius von der neuen Sollbahn und damit vom Objekt entfernt und es so weiträumiger umfährt. Eine Kollision mit anderen Objekten im Operationsgebiet ist durch die geringe Bahnablage im Verhältnis zum gewählten Sicherheitsbereich der Objekte nicht möglich.

Nach einer erfolgreichen Überprüfung mit der Objektliste, kann der Manöverplan durch eine externe *Digitale Seekarte* auf Befahrbarkeit überprüft werden. Eine solche Seekarte enthält ein detailliertes Bodenprofil des Meeres sowie Informationen über gesperrte Gebiete. Erst wenn auch diese zweite Überprüfung erfolgreich war, kann der erstellte Manöverplan abgearbeitet werden.



## 4.6 Abarbeitung des Manöverplanes

Liegt ein überprüfter Manöverplan vor, wird die Führung des Fahrzeuges von der *Reaktiven Steuerung* an die *Manöverplanabarbeitung* übergeben, die ein Bestandteil der *Wegeplanung* ist (siehe Abbildung 3.4). Diese *Manöverplanabarbeitung* generiert unter Verwendung der aktuellen Fahrzeugposition und der Manöverliste Vorgaben für die *Führungsalgorithmen*, wodurch das Fahrzeug der Manöverbahn exakt folgen kann. Dabei müssen zwei Aufgaben erfüllt werden. Zum einen wird in Abhängigkeit von der Fahrzeugposition das aktuell abzufahrende Manöver aus dem Manöverplan ausgewählt. Zum anderen werden für dieses Manöver die Führungsvorgaben in der horizontalen und in der vertikalen Ebene ermittelt. In der horizontalen Ebene erfolgt die Bestimmung der Bahnablage  $e_{Ablage}$  und der Bahnrichtung  $\psi_{Bahn}$  abhängig von der aktuellen Fahrzeugposition zur abzufahrenden Sollbahn des Manövers. Diese Vorgaben werden an den Bahnregler der *Führungsalgorithmen* übergeben (siehe Abschnitt 8.3.3). Die Bestimmung dieser Werte soll nachfolgend für die Strecke und den Kreisbogen vorgestellt werden.

### 4.6.1 Bestimmung der Bahnparameter für die Strecke

Die Bahnrichtung  $\psi_{Bahn}$  der Strecke entspricht der horizontalen Richtung des Streckenmanövers und wird durch die folgende Beziehung beschrieben:

$$\psi_{Bahn} = \text{atan2}(y_{End} - y_{Start}, x_{End} - x_{Start}). \quad (4.15)$$

Die Ermittlung der Bahnablage  $e_{Ablage}$  bestimmt sich aus der y-Koordinate des Entfernungsvektors  $e$  im achsenparallel zur Sollbahn ausgerichteten und in der Endposition  $x_{End}$  gelegenen Koordinatensystems auf Basis von Abbildung 4.29 nach den folgenden Beziehungen [58]:

$$\begin{aligned} e &= R_{\psi_{Bahn}}^T (x_{Fahr} - x_{End}) \\ \begin{bmatrix} l_{Fahr\_End} \\ e_{Ablage} \end{bmatrix} &= \begin{bmatrix} \cos \psi_{Bahn} & \sin \psi_{Bahn} \\ -\sin \psi_{Bahn} & \cos \psi_{Bahn} \end{bmatrix} \begin{bmatrix} x_{Fahr} - x_{End} \\ y_{Fahr} - y_{End} \end{bmatrix} \end{aligned} \quad (4.16)$$

zu:

$$e_{Ablage} = -\sin \psi_{Bahn} (x_{Fahr} - x_{End}) + \cos \psi_{Bahn} (y_{Fahr} - y_{End}). \quad (4.17)$$

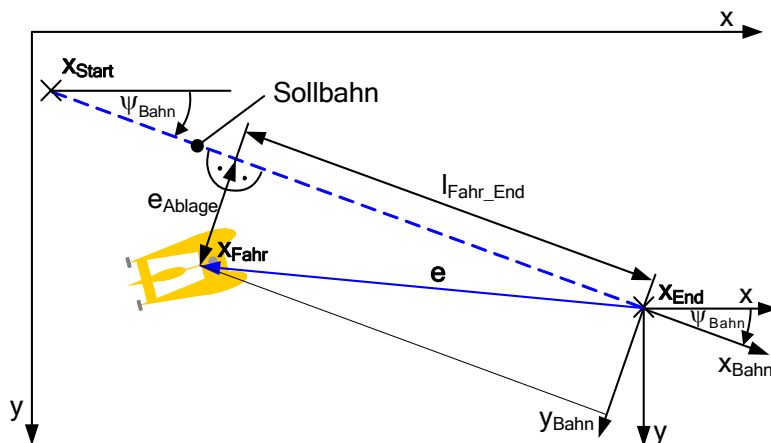


Abbildung 4.29: Ermittlung der Bahnablage beim Streckenmanöver

An dieser Stelle sei angemerkt, dass die Bahnablage negativ definiert ist, wenn sich das Fahrzeug in Bahnrichtung links (backbord) von der Sollbahn und positiv, wenn es sich rechts (steuerbord) davon befindet.

Das aktuelle Streckenmanöver ist beendet, wenn das Fahrzeug eine senkrecht zur Sollbahn verlaufende und durch die Endposition  $\mathbf{x}_{End}$  gehende Linie überfährt. In einem solchen Fall wird die Länge  $l_{Fahr\_End}$  der Gleichung (4.16) positiv.

### 4.6.2 Bestimmung der Bahnparameter für den Kreisbogen

Beim Kreisbogen bestimmt sich die Bahnrichtung  $\psi_{Bahn}$  abhängig vom Richtungswinkel  $\alpha_{Fahr}$ , welcher die Richtung vom Kreisbogenmittelpunkt zum Fahrzeug beschreibt, und der Umfahrrichtung zu:

$$\alpha_{Fahr} = \text{atan2}(y_{Fahr} - y_{Arc}, x_{Fahr} - x_{Arc})$$

$$\psi_{Bahn} = \begin{cases} \alpha_{Fahr} + \frac{\pi}{2} & \text{für Umfahren im Uhrzeigersinn (Drehrichtung positiv)} \\ \alpha_{Fahr} - \frac{\pi}{2} & \text{für Umfahren entgegen dem Uhrzeigersinn (Drehrichtung negativ)} \end{cases} \quad (4.18)$$

Die Bahnablage zum Kreisbogen ergibt sich in Abhängigkeit von der Umfahrrichtung aus dem Kreisbogenradius  $R$  und dem Abstand des Fahrzeuges zum Kreisbogenmittelpunkt  $d_{Fahr\_Arc}$  zu:

$$d_{Fahr\_Arc} = \|\mathbf{x}_{Fahr} - \mathbf{x}_{Arc}\|$$

$$e_{Ablage} = \begin{cases} R - d_{Fahr\_Arc} & \text{für Umfahren im Uhrzeigersinn} \\ d_{Fahr\_Arc} - R & \text{für Umfahren entgegen dem Uhrzeigersinn} \end{cases} \quad (4.19)$$

Das Kreisbogenmanöver gilt als beendet, wenn der aktuelle Fahrzeugrichtungswinkel  $\alpha_{Fahr}$  den Richtungswinkel der Endposition  $\alpha_{End}$  in Abhängigkeit von der Umfahrrichtung unterschritten bzw. überschritten hat.

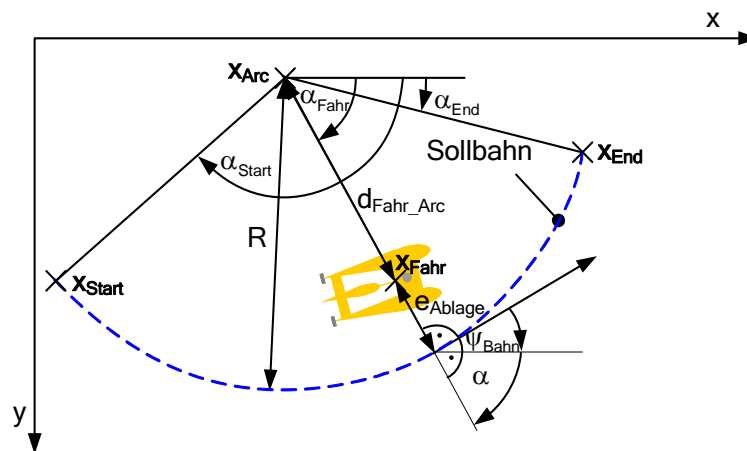


Abbildung 4.30: Ermittlung der Bahnablage beim Kreisbogenmanöver

## 4.7 Test der Verfahren

In diesem Abschnitt werden die entwickelten Algorithmen zur *Wegeplanung* hinsichtlich ihres Laufzeitverhaltens und der generierten Routenpläne untersucht. Dabei werden eigene und in der Literatur beschriebene Testszenarios [81] eingesetzt. Bei den Untersuchungen werden die drei in diesem Kapitel vorgestellten Verfahren zur Erzeugung eines geometrischen Graphen:

- der Sichtbarkeitsgraph (SG),
- der Sektorisierte Sichtbarkeitsgraph (SSG) und
- der Quadtree / Octreegraph (QG / OG)

miteinander verglichen. Weiterhin werden die beiden Suchverfahren zur Bestimmung des kürzesten Weges, der Dijkstra- und der A\*-Algorithmus, hinsichtlich ihres Laufzeitverhaltens abhängig von der Größe des Graphen und der gewählten Startposition untersucht. Die Messbedingungen entsprechen denen des Abschnittes 2.4. Die Tests werden dabei auf einem Rechner mit einem Pentium III 600 MHz Prozessor durchgeführt.

### 4.7.1 Hindernisparcours

Für das erste Testszenario wurde ein Hindernisparcours mit den Abmaßen von 170 m Länge, 84 m Breite, und 50 m Höhe definiert. Innerhalb dieses Operationsgebietes wurden Hindernisse nach dem in Abbildung 4.31 angegebenen Schema platziert. Die Konturen in Abbildung 4.31 entsprechen den um einen Sicherheitsabstand vergrößerten Hindernissen. Tabelle 4.1 zeigt die einzelnen Testbedingungen der vier definierten Tests. Eine Praxisrelevanz von Test 3 und Test 4 ist durch das begrenzte Auffasservermögen eines Sonars nicht gegeben. Durch diese beiden Tests soll die Leistungsfähigkeit der Algorithmen überprüft werden.

Tabelle 4.1: Testbedingungen im Hindernisparcours

Test	Anzahl der Hindernisse	Hindernisnummern
Test 1	5	1-5
Test 2	10	1-10
Test 3	15	1-15
Test 4	20	1-20

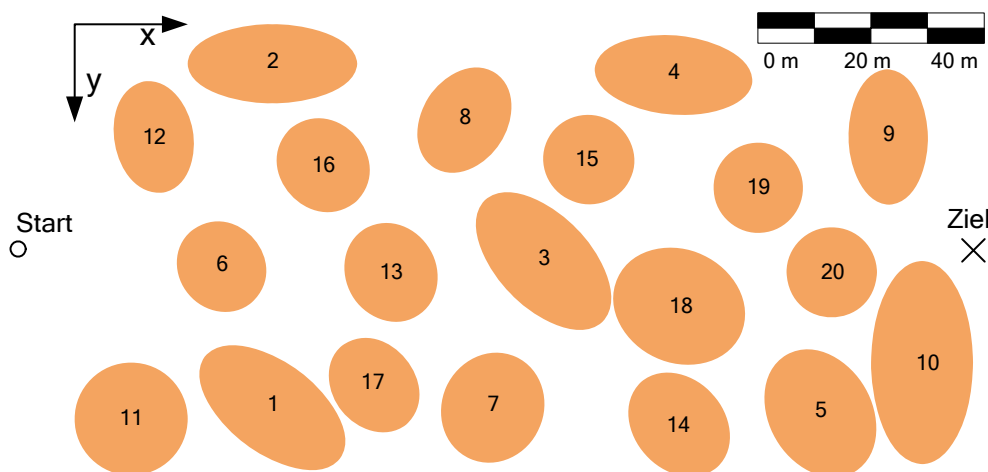


Abbildung 4.31: Hindernisparcours

Die Testreihen wurden für den zwei- und dreidimensionalen Raum durchgeführt. Bei den zweidimensionalen Tests wurden die Hindernisse auf die x-y Ebene projiziert. Die erstellten Graphen enthalten die in Abschnitt 4.3.3 vorgestellte Kostenfunktion. Dies führt zur Generierung eines gerichteten Graphen, da jede Kante in Abhängigkeit von der Fahrtrichtung einen eigenen Kostenwert enthält.

Tabelle 4.2 zeigt die mit den einzelnen Verfahren generierten Kanten und Knoten. Vergleicht man den Sichtbarkeitsgraphen und den Sektorisierten Sichtbarkeitsgraphen, so scheint im zweidimensionalen Raum für wenig Hindernisse (Test 1) der Sichtbarkeitsgraph eine Alternative zum Sektorisierten Sichtbarkeitsgraphen darzustellen. Die Gesamtrechenzeit zur Generierung eines Manöverplanes beträgt dabei trotz ähnlicher Knoten und Kantenanzahl ca. 2 ½ mal mehr als beim Sektorisierten Sichtbarkeitsgraphen. Dies liegt an den durchzuführenden Kollisionsüberprüfungen der möglichen Verbindungen zwischen den einzelnen Knoten und allen Hindernissen im Operationsgebiet laut Gleichung (4.2). Das Verhältnis zwischen Kanten- und Knotenanzahl liegt beim Sektorisierten Sichtbarkeitsgraph und Quadtree- / Octreegraph bei maximal 20 und ist damit sehr gering, was den Einsatz der *Adjazenzliste* gegenüber der *Adjazenzmatrix* zur Haltung des Graphen rechtfertigt.

Tabelle 4.2: Anzahl der Knoten und Kanten der mit den einzelnen Verfahren erzeugten Graphen

Dimension	Test	Sichtbarkeitsgraph		Sektorisierter Sichtbarkeitsgraph		Quadtree / Octreegraph	
		Knoten	Kanten	Knoten	Kanten	Knoten	Kanten
2D	Test 1	62	958	146	918	387	2130
	Test 2	122	2652	176	1116	657	3542
	Test 3	182	4010	212	1504	836	4448
	Test 4	242	4400	258	2118	958	5016
3D	Test 1	182	12834	266	918	1173	17566
	Test 2	362	45396	416	1116	1983	29104
	Test 3	542	87550	447	7956	4259	68336
	Test 4	722	137200	539	11766	4869	76924

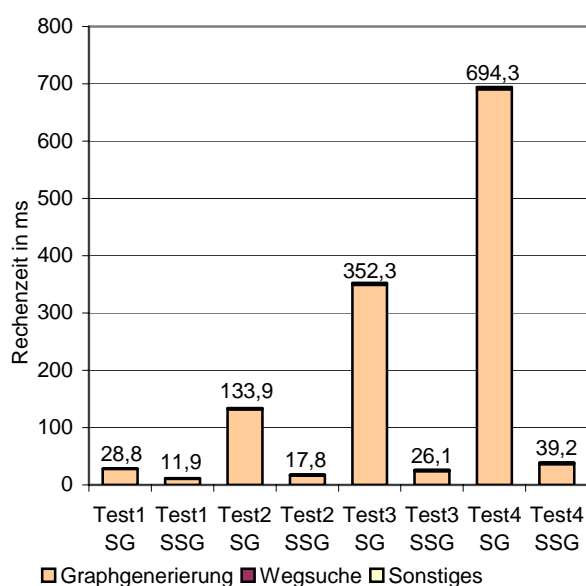


Abbildung 4.32: 2D-Tests Vergleich Sichtbarkeitsgraph (SG) und Sektorisierter Sichtbarkeitsgraph (SSG)

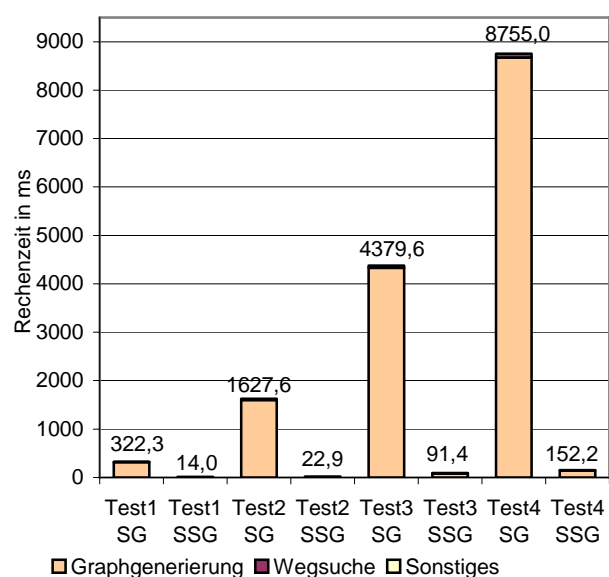


Abbildung 4.33: 3D-Tests Vergleich Sichtbarkeitsgraph (SG) und Sektorisierter Sichtbarkeitsgraph (SSG)

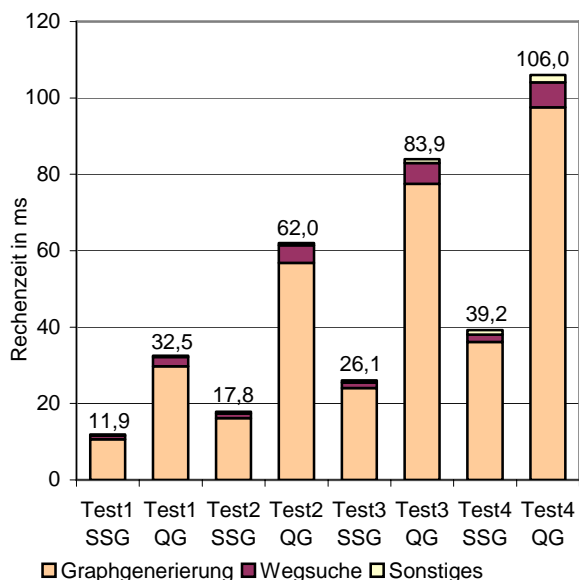


Abbildung 4.34: 2D-Tests Vergleich Sektorisierter Sichtbarkeitsgraph (SG) und Quadtreegraph (QG)

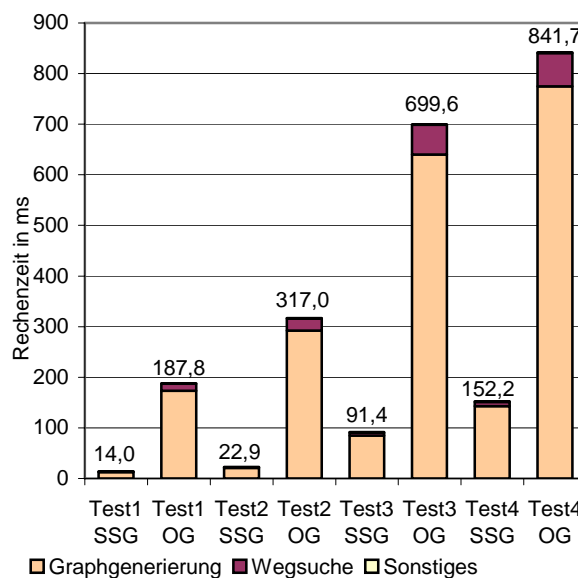


Abbildung 4.35: 3D-Tests Vergleich Sektorisierter Sichtbarkeitsgraph (SG) und Octreegraph (OG)

Die Sektorisierung des Operationsgebietes bei der Erstellung des sektorisierten Sichtbarkeitsgraphen (SSG) führt zu einer deutlichen Verringerung der Rechenzeit gegenüber dem ungeteilten Sichtbarkeitsgraphen (SG) (siehe Abbildung 4.32 und Abbildung 4.33). Dieses Verhältnis der Rechenzeiten des ungeteilten zum sektorisierten Sichtbarkeitsgraphen steigt mit zunehmender Knotenanzahl bzw. bei der Erstellung des Graphen im dreidimensionalen Raum. Der Vergleich von Abbildung 4.34 und Abbildung 4.35 zeigt ein deutliches Anwachsen der Rechenzeit bei den 3D Tests. Die Generierung des Graphen erfordert den größten Zeitaufwand. Die Zeit für die Wegsuche und die Glättung des Routenplanes sowie die Erzeugung des Manöverplanes (in den Abbildungen als *Sonstiges* bezeichnet) nimmt dabei einen verschwindend kleinen Anteil ein. Der Rechenaufwand zur Generierung eines Quadtree- / Octreegraphen liegt um das 3fache im zweidimensionalen Raum und um das 7fache im dreidimensionalen Raum höher als beim Sektorisierten Sichtbarkeitsgraphen. Dies ist durch die notwendige minimale Sektorgröße (1m x 1m x 6m) und die damit verbundene zeitaufwendige Generierung der Datenstrukturen für den 2D- bzw. 3D-Baum sowie die hohe Knoten- und Kantenanzahl gegenüber des Sektorisierten Sichtbarkeitsgraphen begründet. (siehe Tabelle 4.2). Nur durch eine solch feine Diskretisierung können auch *freie* Sektoren zwischen den Hindernissen im vorgegebenen Hindernisparcours generiert werden. Bei allen Testreihen, die einen Sektorisierten Sichtbarkeitsgraphen oder einem Quadtree / Octreegraphen einsetzen, konnte die vorgegebene Zeitvorgabe von einem Sonar-Update (4,5 s) zur Ermittlung eines Manöverplanes sicher eingehalten werden. Für die oben durchgeführten Tests wurde der Dijkstra-Algorithmus zur Bestimmung des kürzesten Weges eingesetzt. Diese Tests wurden mit dem A\*-Algorithmus wiederholt. Als heuristische Funktion zur Abschätzung der Kosten zwischen den einzelnen Knoten und dem Zielpunkt wurde die in Abschnitt 4.3.3 vorgestellte Kostenfunktion eingesetzt. Dabei zeigte sich eine Senkung der Rechenzeit um 1/3 bis zum 8fachen bei Verwendung des A\*-Algorithmus gegenüber des Dijkstra-Algorithmus (siehe Abbildung 4.36 und Abbildung 4.37). Die bei den einzelnen Tests ermittelten Wege waren bei beiden Suchverfahren identisch.

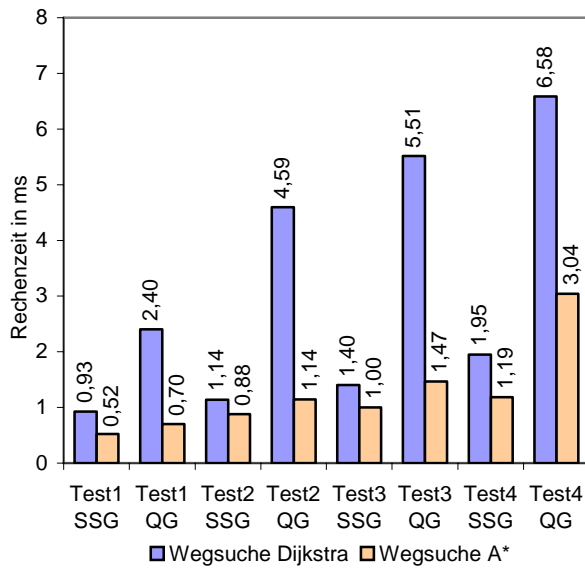


Abbildung 4.36: 2D Tests

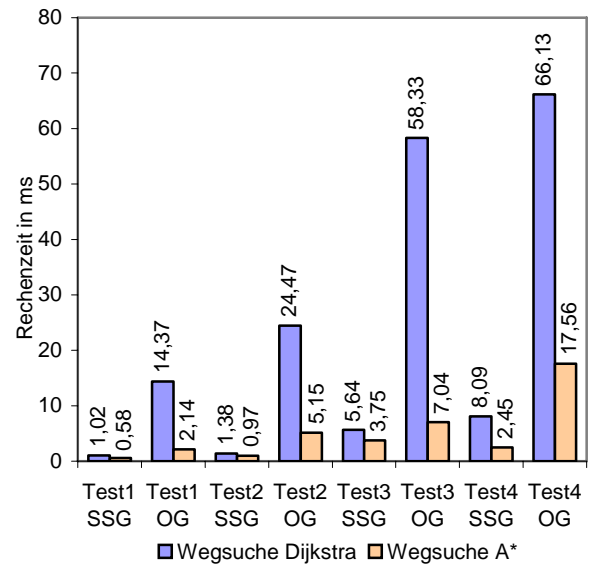


Abbildung 4.37: 3D Tests

Durch die Verwendung des Glättungsalgorithmus konnte die Anzahl der Streckenelemente der gefundenen Routenpläne erheblich gesenkt werden (siehe Tabelle 4.3). Die Glättung führt zu einer Verringerung der Weglängen, die nah am möglichen optimalen Verlauf liegen. Unter optimal wird der mit dem ungeteilten Sichtbarkeitsgraphen erstellte Routenplan angenommen, da dort ein Weg unter Verwendung aller möglicher Verbindungen zwischen den einzelnen Knoten des Graphen ermittelt wird. So zeichnen sich die geglätteten Routenpläne dadurch aus, dass sie nur die notwendigen Richtungswechsel besitzen und die Längen der einzelnen Streckenelemente möglichst groß sind. Die Rechenzeit zum Glätten lag vielfach unter 1 ms. Die Verwendung des Glättungsalgorithmus bei Routenplänen, welche aus den Sektorisierten Sichtbarkeitsgraphen oder den Quadtree- / Octreegraphen erstellt wurden, führte zu ähnlichen Ergebnissen, wie beim ungeteilten Sichtbarkeitsgraphen, jedoch nur in einem Bruchteil der dort benötigten Rechenzeit.

Tabelle 4.3: Weglängen der generierten Routen

Dimension	Test	mit Sicherbar- keits- graphen ermittelter Weg	Sektorisierter Sichtbarkeitsgraph			Quadtree- / Octreegraph		
			Dijkstra & A*			Dijkstra & A*		
			Vor Glätten	Nach Glätten	Rechenzeit zum Glätten in ms	Vor Glätten	Nach Glätten	Rechenzeit zum Glätten in ms
			Kanten- anzahl/ Weglänge in m	Kanten- anzahl/ Weglänge in m		Kanten- anzahl/ Weglänge in m	Kanten- anzahl/ Weglänge in m	
2D	Test 1	3 / 170,32	15 / 172,48	3 / 170,28	0,219	14 / 171,74	2 / 170,46	0,243
	Test 2	4 / 170,34	15 / 176,33	2 / 171,16	0,364	21 / 172,66	3 / 170,65	0,487
	Test 3	5 / 170,43	15 / 175,29	3 / 171,28	0,435	25 / 173,03	4 / 170,98	0,752
	Test 4	9 / 170,95	16 / 175,03	8 / 172,41	0,797	38 / 175,95	7 / 172,79	1,574
3D	Test 1	3 / 169,38	15 / 172,48	3 / 170,28	0,222	14 / 172,08	2 / 170,67	0,243
	Test 2	3 / 169,72	15 / 176,33	2 / 171,16	0,382	21 / 174,19	3 / 170,99	0,482
	Test 3	3 / 169,72	15 / 183,45	8 / 175,05	0,543	25 / 174,83	4 / 171,41	0,746
	Test 4	3 / 170,27	15 / 177,25	11 / 172,22	0,822	12 / 178,34	8 / 176,12	0,563

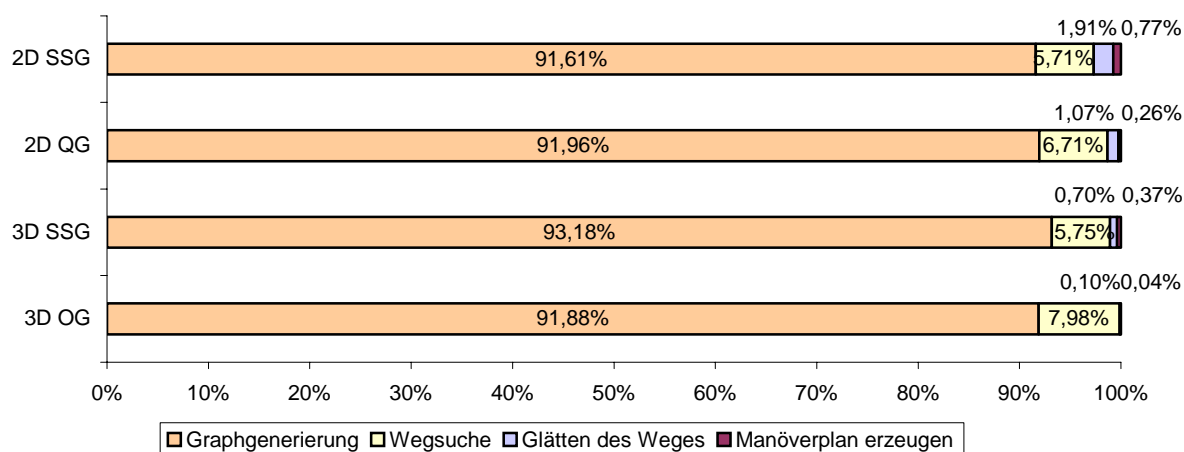


Abbildung 4.38: Prozentuale Anteile der einzelnen Algorithmen zur Ermittlung eines Manöverplanes

Abbildung 4.38 stellt die prozentualen Anteile der Rechenzeiten der einzelnen Algorithmen zur Bestimmung des Manöverplanes dar. Die Werte wurden aus den durchschnittlichen Rechenzeiten der einzelnen Tests für den zwei- und dreidimensionalen Raum unter Verwendung des Sektorisierten Sichtbarkeitsgraphen (SSG) und des Quadtree- / Octreegraphen (QG / OG) ermittelt. Als Suchverfahren wurde der Dijkstra-Algorithmus eingesetzt. Die prozentualen Anteile der einzelnen Algorithmen sind bei allen vier Testbedingungen ähnlich aufgeteilt. Den größten Anteil bildet die Generierung des geometrischen Graphen mit ca. 92 %. Die Wegsuche benötigt nur ca. 6-8 % der Gesamtrechnenzeit. So kann auch die Verwendung des A\*-Algorithmus nicht zu einer signifikanten Senkung der Gesamtrechnenzeit beitragen.

## 4.7.2 Praktisches Testszenario

Für das zweite Testszenario wurde ein in [81] verwendeter Hindernisparcours ausgewählt, der dort zu Untersuchungen bei der Wegeplanung eingesetzt wird. Dabei werden die Suchverfahren der Breitensuche (engl.: Breadth-First) und des A\*-Algorithmus bezüglich ihrer Rechenzeit und der gefundenen Weglängen verglichen. Als Rechner wird eine VAX 8810 eingesetzt, welcher mit einer Taktfrequenz von 22 MHz arbeitet.

Der definierte Hindernisparcours besteht aus 50 Hindernissen und hat eine Ausdehnung von ca. 1800 m x 2200 m. Es wurden 5 Tests definiert, bei denen jeweils von einer anderen Startposition aus, ein Weg zum vorgegebenen Zielpunkt ermittelt werden soll. Abbildung 4.39 zeigt die Positionen und die Ausdehnungen der um die einzelnen Hindernisse definierten Sicherheitsbereiche sowie die Startpositionen und die Endposition. Die Tests finden im zweidimensionalen Raum statt. Das heißt, die Tauchtiefe bleibt beim Abfahren des Manöverplanes konstant. Bei den Tests werden alle Schritte zur Berechnung eines Manöverplanes durchgeführt. Dabei werden die Verfahren zur Generierung eines Sektorisierten Sichtbarkeitsgraphen und eines Quadtreegraphen sowie die Suchverfahren des Dijkstra- und des A\*-Algorithmus eingesetzt und miteinander verglichen.

Die einzelnen Ergebnisse der Tests sind separat für den Sektorisierten Sichtbarkeitsgraphen und den Quadtreegraphen in Tabelle 4.4 und Tabelle 4.5 aufgeführt. Die durchschnittlichen Gesamtrechnenzeiten zur Ermittlung der Manöverpläne liegen bei beiden Graphenverfahren bei 1,3 s, was innerhalb der vorgegebenen Zeitvorgabe von einem Sonar-Update liegt. Dies zeigt deutlich die Leistungsfähigkeit der entwickelten Algorithmen zur Wegeplanung, auch bei einem solch großen und komplexen Operationsgebiet.



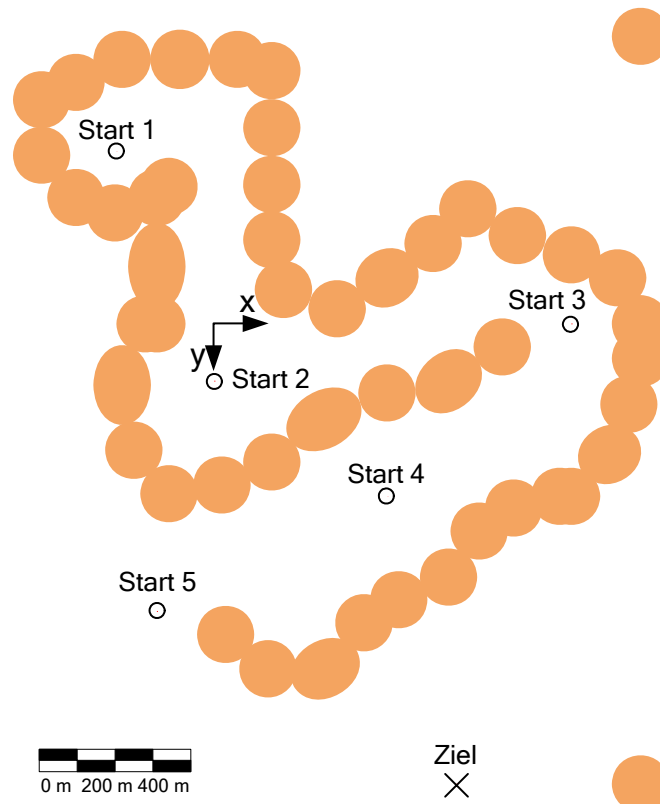


Abbildung 4.39: Hindernisparcours mit Definition der Start- und Endpositionen

In [81] liegen die Rechenzeiten bei Verwendung des A\*-Algorithmus zwischen 142,6 s und 1094,3 s. Durch Umrechnen dieser Werte in die in dieser Arbeit eingesetzte Rechentechnik (22 MHz/600 MHz) liegen die Rechenzeiten somit zwischen 5,2 s und 40,1 s. Dieses Umrechnen ist nur eine grobe Abschätzung, da die Rechenleistung nicht ausschließlich von der Taktfrequenz abhängig ist. Sie zeigt jedoch, dass die in dieser Arbeit entwickelten Algorithmen hinsichtlich der Leistungsfähigkeit mit den in [81] vorgestellten Verfahren sehr gut konkurrieren können.

Tabelle 4.4: Ergebnisse bei Verwendung eines Sektorisierten Sichtbarkeitsgraphen

Test	Suchverfahren	Rechenzeiten in ms				Knoten	Kanten	Vor Glätten	Nach Glätten
		Graph-gene-rierung	Wegsuche	Sonstiges	Gesamt			Kantenanzahl/ Weglänge in m	Kantenanzahl/ Weglänge in m
Test 1	Dijkstra	1252,7	135,18	24,02	1411,9	13837	101094	229 / 3567,21	52 / 3397,41
	A*	1265,46	58,81	24,4	1348,67			229 / 3567,21	51 / 3394,46
Test 2	Dijkstra	1247,58	120,16	21,2	1388,94			197 / 3068,29	34 / 2895,35
	A*	1246,38	57,32	21,72	1325,42			197 / 3068,29	34 / 2895,35
Test 3	Dijkstra	1233,47	121,49	14,92	1369,88			134 / 2248,73	14 / 2108,36
	A*	1234,17	45,74	15,22	1295,12			134 / 2248,73	14 / 2108,36
Test 4	Dijkstra	1244,98	122,84	11,14	1378,96			96 / 1552,7	17 / 1444,36
	A*	1288,96	29,21	11,17	1269,36			96 / 1552,7	17 / 1444,36
Test 5	Dijkstra	1235,48	123,34	6,13	1364,94			60 / 991,646	5 / 923,32
	A*	1241,38	13,53	6,02	1260,93			60 / 991,646	5 / 923,32
Durchschnitt	Dijkstra	<b>1242,84</b>	<b>124,60</b>	<b>15,48</b>	<b>1382,92</b>				
	A*	<b>1243,27</b>	<b>40,92</b>	<b>15,71</b>	<b>1299,90</b>				



Tabelle 4.5: Ergebnisse bei Verwendung eines Quadtreegraphen

Test	Suchverfahren	Rechenzeiten in ms				Knoten	Kanten	Vor Glätten	Nach Glätten
		Graph-gene-rierung	Wegsuche	Sonstiges	Gesamt			Kantenanzahl/ Weglänge in m	Kantenanzahl/ Weglänge in m
Test 1	Dijkstra	1264,19	85,79	19,82	1369,8	11502	60632	247 / 3673,53	16 / 3486,87
	A*	1269,78	63,01	20,63	1353,42			254 / 3673,53	15 / 3446,16
Test 2	Dijkstra	1267,83	86,62	12,17	1366,62			146 / 3132,47	10 / 2984,28
	A*	1267,16	62,31	13,09	1342,56			154 / 3132,47	10 / 2943,28
Test 3	Dijkstra	1277,24	88,11	5,98	1371,33			68 / 2298,19	6 / 2188,73
	A*	1273,67	44,06	6,76	1324,49			75 / 2298,19	6 / 2147,73
Test 4	Dijkstra	1266,28	88,8	2,96	1358,04			31 / 1626,77	4 / 1476,96
	A*	1288,63	22,88	3,45	1314,96			38 / 1626,77	4 / 1476,96
Test 5	Dijkstra	1272,91	90,28	1,4	1364,59			14 / 1098,63	2 / 969,176
	A*	1275,11	9,44	1,87	1286,42			21 / 1098,63	2 / 932,9
Durchschnitt	Dijkstra	<b>1269,69</b>	<b>87,92</b>	<b>8,466</b>	<b>1366,08</b>				
	A*	<b>1274,87</b>	<b>40,34</b>	<b>9,16</b>	<b>1324,37</b>				

Das Glätten der Routenpläne führt bei beiden generierten Graphentypen zu einer signifikanten Abnahme der Streckenelemente bei gleichzeitiger Verkürzung der Weglängen. Die Rechenzeiten zur Ermittlung des kürzesten Weges sind bei Verwendung des Dijkstra-Algorithmus bei allen fünf durchgeführten Tests nahezu gleich (siehe Abbildung 4.40). Die Rechenzeit beim Quadtreegraphen liegt dabei um 40 % niedriger als beim Sektorisierten Sichtbarkeitsgraphen, was direkt mit der verwendeten Kantenanzahl der beiden Graphen zusammenhängt. Bei Verwendung des A\*- Algorithmus korreliert die Rechenzeit mit der direkten Entfernung zwischen Start- und Zielpunkt. Dies liegt in der Arbeitsweise des A\*-Algorithmus begründet, der nur eine Teilmenge aller Knoten bei der Wegeplanung einbezieht, und zwar bevorzugt die Knoten mit einer kurzen Entfernung zum Zielpunkt. Im Anhang sind die generierten Graphen mit den ermittelten Wegen für den Sektorisierten Sichtbarkeitsgraphen und den Quadtreegraphen abgebildet. Beide ermittelten Wege zeigen einen ähnlichen Verlauf, der dadurch gekennzeichnet ist, dass die einzelnen Streckenelemente als Tangenten um die Hindernisse

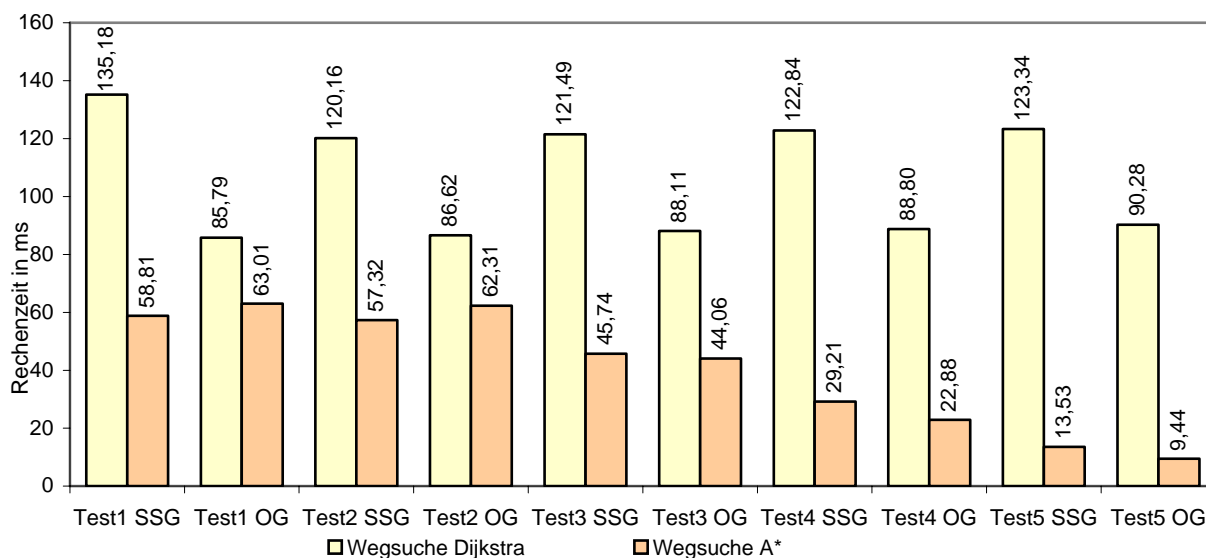


Abbildung 4.40: Rechenzeiten bei der Wegsuche bei Verwendung des Dijkstra- und des A\*-Algorithmus

herumgeführt werden. Beim Sektorisierten Sichtbarkeitsgraphen ähnelt das entstandene Kantennetz der gitterbasierten Darstellung des Abschnittes 4.1.2.1. Dies liegt daran, dass die Sektoren eine Ausdehnung von 15 m x 15 m haben. Dieser Wert hat sich bei den in [86] durchgeführten Untersuchungen bezüglich der entstandenen Kantenanzahl und der Rechenzeit für die Graphengenerierung als optimal erwiesen. Die Untersuchungen wurden mit den im Abschnitt 4.7.1 verwendeten Hindernisparcours durchgeführt. Die Objektgrößen sowie die Ausdehnung des Operationsgebietes sind bei dem in diesem Abschnitt untersuchten Szenario um eine Dekade größer. So entstehen sehr viele freie Sektoren in deren Mittelpunkten Knoten definiert werden (siehe Abschnitt 4.3.2.1). Eine Anpassung der Sektorgröße würde zu einer Senkung der Gesamtrechenzeit führen. Es zeigt sich, dass aber auch mit der gewählten Sektorgröße die Laufzeitanforderung erfüllt wird.

## 4.8 Zusammenfassung

Die Aufgabe der Wegeplanung besteht in der Generierung eines zeit- bzw. wegoptimalen Manöverplanes unter Einhaltung der vorgegebenen Zeitvorgabe sowie dessen Abarbeitung. Durch die im Abschnitt 4.2.1 definierten Anforderungen wurden zur Generierung des Manöverplanes graphenbasierte Verfahren favorisiert. Diese Verfahren ermöglichen es, einen optimalen Weg nach definierten Vorgaben in einer kalkulierbaren Zeit zu ermitteln. Für die Erstellung eines geometrischen Graphen wurden in diesem Kapitel die entwickelten Verfahren des Sektorisierten Sichtbarkeitsgraphen und des Quadtree- / Octreegraphen vorgestellt. Die Auswahl dieser Verfahren erfolgte auf Grundlage der vorgegebenen geometrischen Form zur Beschreibung der Objekte (elliptischer Zylinder) und der Forderung nach effizienten Algorithmen, die mit den zur Verfügung stehenden Prozessor- und Speicherressourcen auskommen. Die Einbeziehung der Seeströmung in die Kostenfunktion ermöglicht es, Gebiete mit starker Seeströmung von der Routenplanung auszuschließen. Für die Wegsuche wurde die Boost Graph Library [16, 20, 138] eingesetzt. Sie beinhaltet eine Bibliothek mit erprobten Datenstrukturen für Graphen und Suchalgorithmen und unterstützt eine Vielzahl von C++ Compilern. Als Suchverfahren kamen der Dijkstra- und der A\*-Algorithmus zum Einsatz. Da die in dieser Arbeit entwickelten Verfahren zur Graphenerzeugung keine vollständigen Graphen erzeugen, zeigen die ermittelten Routen einen stufen- bzw. schlangenförmigen Verlauf. In Abschnitt 4.4.2 wird ein neu entwickeltes Verfahren vorgestellt, welches diese Routenpläne unter Einhaltung der Kollisionsfreiheit mit den Objekten glättet. Die geglätteten Routen sind den aus einem vollständigen Graphen erzeugten Routen sehr ähnlich. Die notwendige Rechenzeit zur Erzeugung dieser Routen von der Graphengenerierung, über die Wegsuche, bis hin zur Glättung entspricht jedoch nur einem Bruchteil der Rechenzeit, die zur Routenerzeugung bei einem vollständigen Graphen benötigt wird.

Die durchgeführten Laufzeituntersuchungen zeigen die Leistungsfähigkeit der neu entwickelten Algorithmen. Die Gesamtrechenzeiten zur Erzeugung der Manöverpläne bei den einzelnen Testszenarios lagen bei beiden geometrischen Graphen innerhalb der einzuhaltenden Zeitvorgabe. Der Hauptteil der Rechenzeit mit 92 % wird zur Generierung der Graphen benötigt. Eine Analyse der generierten Routenpläne zeigt, dass die Streckenabschnitte als Tangenten an den Objekten vorbeiführen. Dieser Verlauf wird bei dem im Abschnitt 4.1.2.4 vorgestellten Kreistangentenansatz genutzt. Deshalb wird der Einsatz dieses Verfahrens in kommenden Arbeiten mit dem Ziel, die Rechenzeit zur Generierung des Graphen weiter zu senken, untersucht.

## 5 Reaktive Steuerung

Dieses Kapitel behandelt den reaktiven Teil des Kollisionsvermeidungssystems. Ausgehend von einer Beschreibung des Aufgabenbereiches und den Anforderungen an die *Reaktive Steuerung* im Modul *FIS* erfolgt eine Vorstellung von bekannten und praxiserprobten Verfahren. Einige dieser Ansätze wurden im Rahmen dieser Arbeit auf ihre Einsatzmöglichkeit zur Führung eines AUVs untersucht. Aus den gesammelten Erkenntnissen der untersuchten Verfahren und ihrer begrenzten Einsatzfähigkeit für die Reaktive Steuerung wurde ein neues Verfahren auf der Basis harmonischer Dipol-Potentiale entwickelt [71]. Es wird in Abschnitt 5.3 detailliert vorgestellt. Anschließend finden ein Vergleich des neuen Verfahrens mit dem Basisverfahren sowie eine Überprüfung seiner Echtzeit- und Leistungsfähigkeit statt.

### 5.1 Aufgabenbereich und Anforderungen der Reaktiven Steuerung

Innerhalb des Moduls *Fahrzeugführung in Sondersituationen* kann die *Reaktive Steuerung* in zwei Betriebsarten eingesetzt werden. Im ersten Fall ist sie ein Teil des *Kollisionsvermeidungssystems*, welches in Kapitel 3 beschrieben wird. Sie ist während der Erzeugung und Prüfung der Route in der Wegeplanung aktiv. Weiterhin löst sie diese bei plötzlich auftauchenden Hindernissen oder einer zu großen Trackablage des Fahrzeuges zur generierten Route ab. Sie übernimmt dabei die Fahrzeugführung, während die *Wegeplanung* die Route unter Verwendung der neuen Objektinformationen und der aktuellen Fahrzeugposition modifiziert bzw. neu erstellt.

Ein anderer Einsatzfall ist bei der Bereitstellung „ungenauer“ Objektdaten durch das Modul *Objekterkennung* gegeben. Dieser Fall wurde zu Projektbeginn als „Worst Case“ Szenario angenommen, da es zu diesem Zeitpunkt noch keine Aussage über die Leistungsfähigkeit des zu entwickelnden Objekterkennungssystems gab. Unter „ungenau“ ist ein stochastisches Variieren der Abmaße und der Lage der Objekte infolge der begrenzten Reichweite der Aufklärungs-Sensorik (z. B. Forward Looking Sonar) sowie der entfernungsabhängigen Detektionsgüte zu verstehen. In solchen Fällen ist ein Ausweichverfahren gefordert, welches eine hohe Robustheit gegenüber solchen Variationen besitzt. Diese Forderung wird durch die Reaktive Steuerung erfüllt, da sie die Steuerkommandos (Sollkurs, -tiefe und -geschwindigkeit) für den Autopiloten nur auf der Basis der aktuellen Lage der Objekte zum Fahrzeug generiert.

Das eingesetzte Verfahren soll die Manövrierfähigkeit des Fahrzeuges berücksichtigen und die Art der bereitgestellten Objektinformationen in Form von elliptischen Zylindern unterstützen. Eine Verwendung der Sonarrohdaten für die reaktive Steuerung ist durch ihre große Updatezeit von 4,5 s und mehr (siehe Abschnitt 1.2.2) nicht möglich.

## 5.2 Übersicht über bekannte Realisierungen von reaktiven Ansätzen

Dieser Abschnitt gibt einen Überblick über bekannte, praxiserprobte Verfahren der Reaktiven Steuerung. Bei der Auswahl der Verfahren wurde auf eine mögliche Einsatzfähigkeit im Modul *FIS*, unter Beachtung der im Abschnitt 1.2 beschriebenen Anforderungen, Wert gelegt.

### 5.2.1 Künstliche Potentialfelder

Die Verwendung von virtuellen Potentialfeldern zur Steuerung eines Roboters bei der Kollisionsvermeidung ist ein verbreiteter und vielfach modifizierter Ansatz. Er zeichnet sich durch seine Einfachheit und Plausibilität aus. Die Idee besteht darin, dass alle Objekte bzw. Hindernisse auf den Roboter abstoßende Kräfte ausüben. Der anzufahrende Zielpunkt dagegen übt eine anziehende Kraft auf den Roboter aus. Die abstoßenden (repulsiven) Kräfte werden durch Potentiale hervorgerufen, die sich nach der folgenden Formel bestimmen lassen [90, 102]:

$$U_{rep}^i(\mathbf{q}) = \begin{cases} \frac{1}{2} \eta \left( \frac{1}{p_i} - \frac{1}{p_o} \right)^2 & \text{wenn } p_i \leq p_o \\ 0 & \text{sonst} \end{cases}$$

mit  $p_i = \|\mathbf{q} - \mathbf{q}_i\|$  (5.1)  
 $\mathbf{q}$  aktuelle Roboterposition  
 $\mathbf{q}_i$  Mittelpunkt des i-ten Objektes  
 $p_o$  maximaler Einflussbereich des Potentials  
 $\eta$  empirischer Faktor

Die repulsive Kraft bestimmt sich durch Bildung des Gradienten  $\mathbf{F}_{rep}^i(\mathbf{q}) = -\text{grad} U_{rep}^i(\mathbf{q})$  zu:

$$\mathbf{F}_{rep}^i(\mathbf{q}) = -\eta \left( \frac{1}{p_i} - \frac{1}{p_o} \right) \frac{1}{p_i^2} \begin{pmatrix} \cos \varphi_i \\ \sin \varphi_i \end{pmatrix}. \quad (5.2)$$

Für das anziehende (attraktive) Potential wird eine konische Potentialfunktion [74, 147]:

$$U_{att}(\mathbf{q}) = \|\mathbf{q} - \mathbf{q}_z\| \quad (5.3)$$

gewählt. Diese Funktion führt, im Gegensatz zu der in [90, 102] verwendeten parabolischen Potentialfunktion, zu einer konstanten Anziehungskraft  $\mathbf{F}_{att}$  unabhängig von der Entfernung zum Zielpunkt  $\mathbf{q}_z$ :

$$\mathbf{F}_{att}(\mathbf{q}) = -\text{grad} U_{att}(\mathbf{q}) = \begin{pmatrix} \cos \varphi_z \\ \sin \varphi_z \end{pmatrix}. \quad (5.4)$$

Durch Addition aller repulsiven Potentiale  $U_{rep}^i$  und des attraktiven Potentials  $U_{att}$  erhält man ein Potentialfeld  $U(\mathbf{q})$ , dessen negativen Gradienten der Roboter folgt. Abbildung 5.1 zeigt ein so entstandenes Potentialgebirge mit dem in Abbildung 5.2 verwendeten Objektszenario.

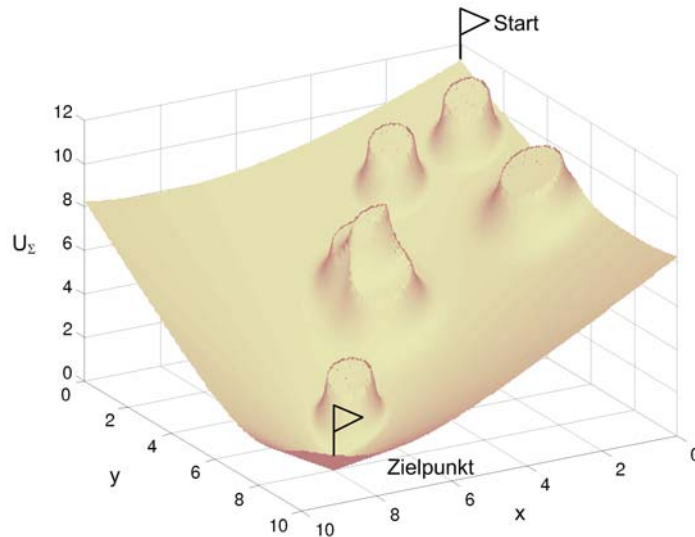


Abbildung 5.1: Potentialgebirge

Der Gradient entspricht der auf den Roboter wirkenden Gesamtkraft  $\mathbf{F}$ , die sich durch Addition der einzelnen Kräfte der Gleichungen (5.2) und (5.4) zu:

$$\mathbf{F} = \mathbf{F}_{att} + \sum_{i=1}^n \mathbf{F}_{rep}^i . \quad (5.5)$$

bestimmt. Abbildung 5.2 zeigt den abgefahrenen Weg des Roboters durch einen Hindernisparcours unter Verwendung der Potentialfeldmethode. Dieser Weg weist plötzliche Richtungswechsel auf, denen ein Unterwasserfahrzeug nur mit Verzögerung folgen kann. Ein anderes Problem ist das Festsitzen in lokalen Minima (siehe Abbildung 5.3). Dies erfolgt, wenn sich die anziehende Kraft und die abstoßenden Kräfte aufheben. Weitere Nachteile dieses Verfahrens ist die Abhängigkeit der ausgeübten Kraft von der Objektgröße [19], das Blockieren bei Engstellen und das Oszillieren beim Durchfahren von engen Gängen [19, 71, 74]. Eine Übersicht über die Probleme beim Einsatz von Potentialfeld-Methoden wird in [97] vorgestellt.

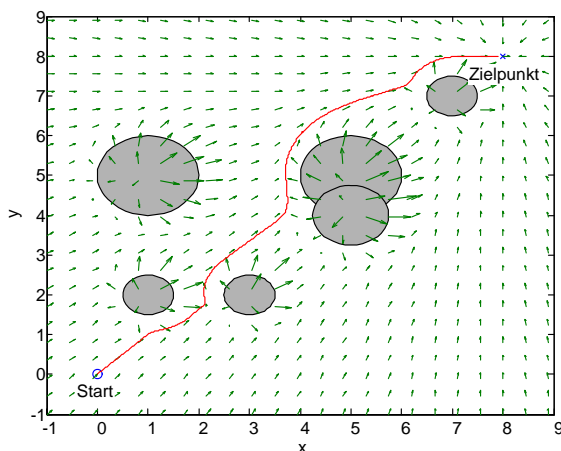


Abbildung 5.2: Fahrtrajektorie

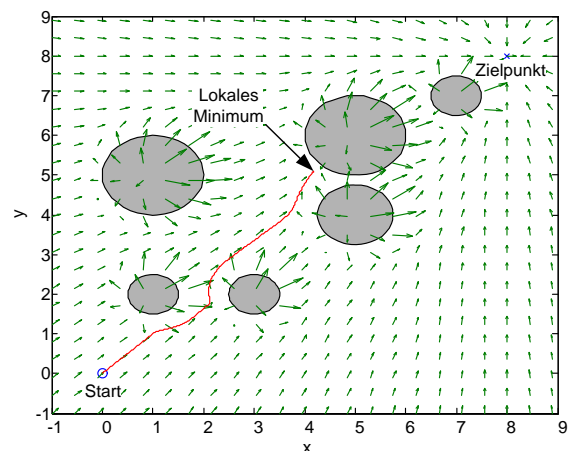


Abbildung 5.3: Festsitzen im lokalen Minimum

## 5.2.2 Fuzzybasierte Steuerung

Der Einsatz eines Fuzzy-Systems für die Reaktive Steuerung der Kollisionsvermeidung bietet sich durch die einfache Abbildung des menschlichen Wissens und Verhaltens in linguistischen Regeln an. So kann durch das Fuzzy-System bei einer vorhandenen Kollisionssituation eine dem menschlichen Verhalten äquivalente Aktion ausgeführt werden. Ein solches Fuzzy-System generiert in Abhängigkeit von den am Eingang anliegenden Merkmalen und unter Verwendung der hinterlegten Regelbasis entsprechende Entscheidungen. Systemanalytisch besitzt es ein nichtlineares, statisches Übertragungsverhalten [93]. In [134] wird die Fuzzy-Logik für ein Kollisionsvermeidungssystem des Miniroboters Khepera eingesetzt. Die einzigen Informationen über die Umwelt erhält der Roboter durch 8 IR-Sensoren. Diese werden zu vier Sektorgruppen (*links*, *rechts*, *vorn*, *hinten*) zusammengefasst und bilden die Eingänge des Fuzzy-Systems. In Abhängigkeit von der Entfernung der in den einzelnen Sektorgruppen detektierten Objekte werden Regeln aktiviert, die eine Entscheidung zur Bewegung des Roboters bilden. In [67] kommt die Fuzzy-Logik in einer Drei-Ebenen Architektur für ein Autonomes Unterwasserfahrzeug, beginnend bei der Kollisionserkennung über die Führungsregelungen bis zu den Lageregelungen, zum Einsatz. Der Fuzzy-Ansatz wird hier wegen seiner guten Robustheit bezüglich unsicherer Sonardaten und einer ungenauen Navigation favorisiert. Im Rahmen des BMBF-Projektes „DeepC“ [55] wurden Maschinelle Lernverfahren zur Generierung von Regelwerken für die Kollisionsvermeidung untersucht. Dabei werden für den verwendeten ID3-Algorithmus Lerndaten aus den Testfahrten eines Operators in einer virtuellen Realität bei unterschiedlichen Umweltsituationen aufgezeichnet [89]. Die Testumgebung zur Aufzeichnung dieser Lerndaten ist in Abbildung 5.4 dargestellt. In dieser Testumgebung steuert der Operator das Fahrzeug über einen Joystick. Die simulative Führung des Fahrzeuges sowie die Darstellung der Instrumente und des Sonars erfolgt durch MATLAB®/Simulink® [142]. Für die virtuelle Realität wird KISMET eingesetzt [57].

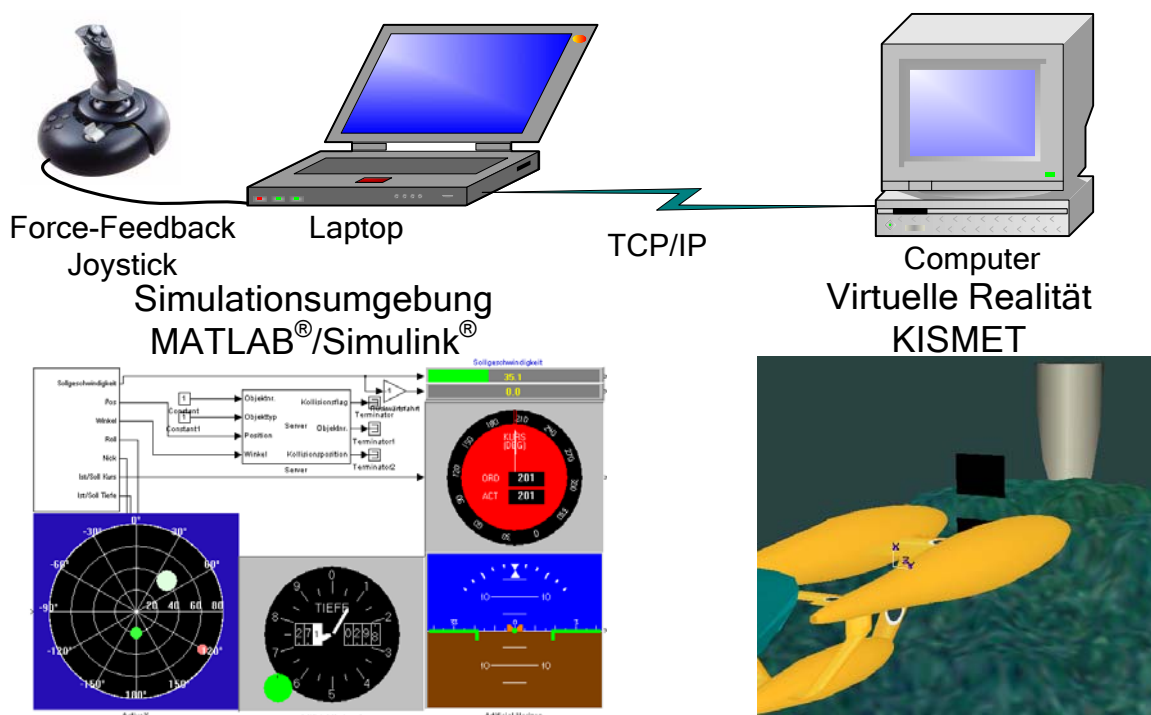


Abbildung 5.4: Testumgebung zur Aufnahme der Lerndaten in der Virtuellen Realität

Für die Arbeit mit dem Fuzzy-System war es notwendig, das durch die einzelnen Hindernisdaten (Position und Form der elliptischen Zylinder) beschriebene Weltmodell (siehe Abschnitt 1.3.1) in eine für das Fuzzy-System erfassbare bzw. bearbeitbare Form zu bringen. Bei einer direkten Eingabe der Objektdaten mit Position, Größe und Lage in das Fuzzy-System können keine Regeln aufgestellt werden, die das Fahrzeug vorbei an den Hindernissen zum Zielpunkt führen. Dies ist verständlich, wenn man sich das menschliche Handlungsschema beim Lösen dieser Aufgabe betrachtet. Hier wird das Hindernisszenario ausgewertet, um einen Weg zum Zielpunkt zu finden, wobei die Hindernisse in einem vorgegebenen Sicherheitsabstand zu umfahren sind. Dabei gehen im menschlichen Denkprozess mehrere geometrische Zwischenschritte vor, wie z.B. das Bestimmen von Abstand und Tangente zum Hindernis oder das Lokalisieren freier Durchfahrten zwischen den Hindernissen. Eine solche Aufbereitung kann ein Fuzzy-System, das nur aus Wenn-Dann-Regeln besteht, nicht durchführen. Deshalb müssen diese vorbereiteten Informationen dem Fuzzy-System in einer geeigneten Form zur Verfügung gestellt werden. Dazu werden um das Fahrzeug  $n$  Sektoren definiert, die als Wert die kürzeste Hindernisentfernung besitzen. Dies ist vergleichbar mit den IR-Sensoren des Miniroboters Khepera [134]. Durch diesen Schritt findet sozusagen eine Rücktransformation des Weltmodells in die Sensorebene statt. In einigen Veröffentlichungen [67] wird als wesentlicher Vorteil der Fuzzy-Technologie ihre Robustheit und die damit verbundene Verwendung der Sensorrohdaten als Eingangsgröße für das Fuzzy-System genannt. Für das in dieser Arbeit eingesetzte Sonar mit einer Updaterate von ca. 4,5 s (siehe Abschnitt 1.2.2) und der geforderten Updatezeit des Moduls *FIS* von 100 ms ist die Verwendung der Sonarrohdaten nicht möglich. Dies würde dazu führen, dass nach Erhalt der Sonarrohdaten 4,5 s lang mit den gleichen Eingangsgrößen für das Fuzzy-System gearbeitet wird. Eine in diesem Zeitraum durchgeführte Bewegung des Fahrzeuges und eine damit verbundene Veränderung seiner Lage zu den Hindernissen findet so keine Berücksichtigung bei der Generierung der Steuerkommandos. Da die Reaktive Steuerung einen geschlossenen Führungskreis darstellt, ist diese Problematik in Analogie zu Regelkreisen mit einer zu großen Abtastzeit zu sehen.

Abbildung 5.5 und Abbildung 5.6 zeigen Fahrtrajektorien des Fuzzy-Systems zur Kollisionsvermeidung in einer simulativen Erprobung unter MATLAB®. Das reaktive Verhalten bei einer Hindernisumfahrung ist deutlich zu erkennen. Es ähnelt den im Abschnitt 5.2.1 dargestellten Trajektorien.

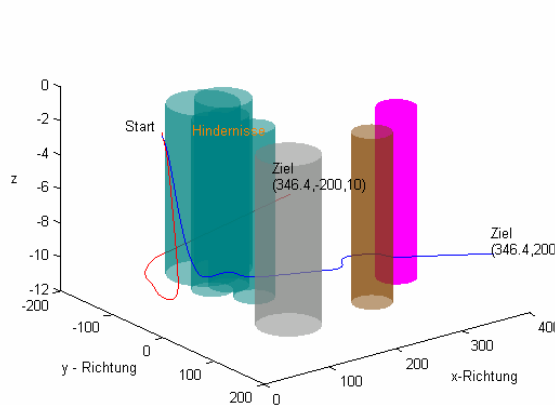


Abbildung 5.5: Fahrtrajektorien 3D Ansicht [55]

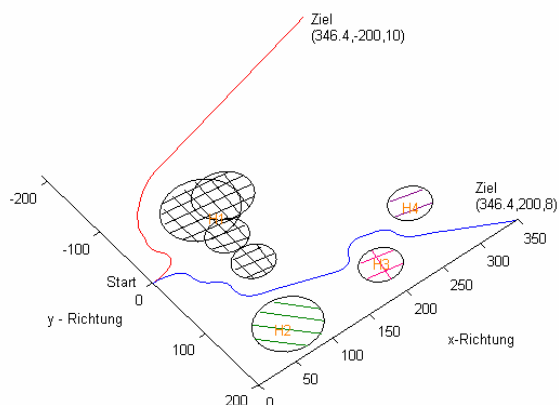


Abbildung 5.6: Fahrtrajektorien [55]



### 5.2.3 Vektorfeld-Histogramm

In [21] wird ein Verfahren zur Kollisionsvermeidung beschrieben, das die Probleme der im Abschnitt 5.2.1 vorgestellten künstlichen Potentialfelder (lokale Minima, Blockieren bei Engstellen) kompensiert. Diese Methode wird als Vektorfeld-Histogramm bezeichnet und ist eine Weiterentwicklung der Virtuellen Kräftefeld-Methode. Beide Verfahren bilden das Operationsgebiet in einem Gitter ab. Jede einzelne Zelle dieses Gitters enthält einen Wahrscheinlichkeitswert, welcher der Anzahl der in dieser Zelle gemessenen Hindernisdetektionen entspricht. So baut sich, während der Roboter durch das Operationsgebiet fährt, ein Histogramm-Gitter auf, bei dem die Zellen mit einem hohen Wahrscheinlichkeitswert den Gebieten mit einer wirklichen Hindernisbelegung entsprechen. Diese Vorgehensweise bietet sich bei der Verwendung von ungenauen und gestörten Sensordaten an.

In der zweiten Stufe der Vektorfeld-Histogramm-Methode werden um den Roboter Sektoren definiert. Für jeden Sektor wird ein Wert berechnet, der sich aus den Wahrscheinlichkeitswerten der in diesem Sektor liegenden Zellen sowie deren Abstand zum Roboter bestimmt. So entsteht ein Polar-Histogramm, welches die Wahrscheinlichkeitswerte der einzelnen Sektoren in Abhängigkeit der Sektorrichtung darstellt. Ein solches Histogramm besteht aus „Bergen“ und „Tälern“, wobei die „Berge“ die Gebiete mit einer hohen Hinderniswahrscheinlichkeit und die „Täler“ die befahrbaren Bereiche klassifizieren (siehe Abbildung 5.7). Die Fahrtrichtung für den Roboter bestimmt sich aus dem zur Zielrichtung nächstgelegenen „Tal“. Ein Nachteil dieses Verfahrens besteht in der aufwendigen Erzeugung des Histogramm-Gitters und der Generierung des Polar-Histogramms. In [19] wird deshalb auf eine Erzeugung des Polar-Histogramms über das Histogramm-Gitter verzichtet. Hier wird die Sensorinformation über die Richtung und den Abstand zum detektierten Objekt durch mehrfaches Messen verifiziert und direkt den entsprechenden Sektoren zugeordnet. So entsteht ein binäres Polar-Diagramm, das die Aussage über die Belegtheit der einzelnen Sektoren enthält.

Ein ähnlicher Ansatz zur Nutzung von befahrbaren Sektoren wird bei der „Nearness-Diagram“-Methode verwendet [110, 111]. Dabei wird ein Sichtkreis in einzelne Sektoren unterteilt, die mit Hindernissen belegt oder befahrbar sind. Anschließend wird ein freier Sektor ausgewählt, um in Abhängigkeit der aktiven Sicherheitssituation die Steuerkommandos zu berechnen. Durch das Hin- und Herschalten zwischen zwei Situationen kann es dabei zu Unstetigkeiten und zum Oszillieren kommen [74]. Diese Methode wurde in [74] als Reaktive Hindernisvermeidung für einen mobilen Roboter in einer Robocup-Umgebung favorisiert.

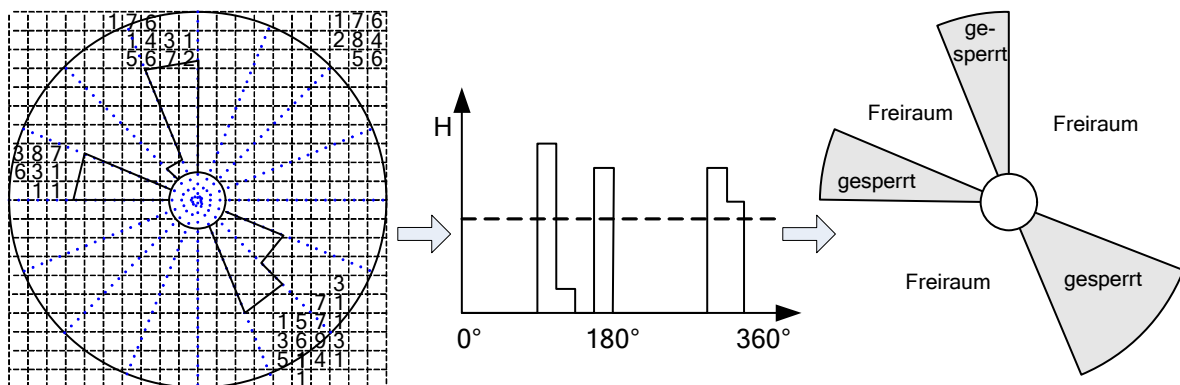


Abbildung 5.7 Dreistufige Datenrepräsentation beim Vektorfeld-Histogramm [19]



### 5.2.4 Harmonische Dipol-Potentiale

In [69, 70, 71, 72] wird ein Kollisionsvermeidungsverfahren auf Basis harmonischer Dipol-Potentiale vorgestellt, welches die in [97] beschriebenen Probleme künstlicher Potentialfelder löst. Solche Probleme sind das „Festsetzen“ des Fahrzeuges in lokalen Minima, die oszillierende Bewegung in engen Durchfahrten sowie das Sperren von möglichen Routen durch das resultierende abstoßende Kraftfeld am Eingang einer Durchfahrt [72]. Die Idee des Verfahrens besteht in der Verwendung von Bahnlinien, die von jeder beliebigen Position des Operationsgebietes einen Weg zum Zielpunkt aufzeigen. Der Sollkurs des Fahrzeuges wird durch Bildung des Bahnanstiegs (Gradienten  $\mathbf{G}$ ) an der aktuellen Fahrzeugposition bestimmt. So kann das Fahrzeug von seiner aktuellen Position aus die aktuelle Bahnlinie bzw. ihren Gradienten verwenden, um vorbei an den Hindernissen zum Zielpunkt geführt zu werden. Diese Bahnlinien werden durch Bildung der Gradientenlinien eines Dipols erzeugt. Dabei wird ein Dipol-Potential zwischen einem Hindernis und dem Zielpunkt definiert, welches dadurch charakterisiert ist, dass seine Gradientenlinien den Sicherheitskreis des Hindernisses nicht schneiden und alle zum Zielpunkt führen. Dieses Potentialfeld wird nur für das dem Fahrzeug nächstgelegene Hindernis berechnet. So fährt das Fahrzeug, geführt durch die Gradientenlinien des aktuell verwendeten Dipol-Potentials von Hindernis zu Hindernis, vergleichbar mit einem Wanderer, der über Steine springt, um einen Fluss zu überqueren. In den nachfolgenden Ausführungen soll die Herleitung der Gradientenlinien für die harmonischen Dipol-Potentiale dargelegt werden, wobei das Hindernis durch einen Kreis beschrieben wird.

Fasst man den Kreis des Hindernisses als positive Ladung und den anzufahrenden Zielpunkt als negative Punktladung auf, so bildet diese Struktur einen Dipol, dessen Feldlinien von der positiven zur negativen Ladung führen. Diese Feldlinien bilden die möglichen Fahrtrouten eines Fahrzeuges von seiner aktuellen Position aus zum Zielpunkt. Für ein elektrostatisches Feld gelten die folgenden Beziehungen zwischen der Feldstärke  $\mathbf{E}$  und dem Potential  $\varphi$ :

$$\mathbf{E} = -\text{grad } \varphi, \quad (5.6)$$

wobei sich  $\varphi$  für den 2-dimensionalen Raum, in dessen Zentrum sich die Ladung  $Q$  befindet, zu:

$$\varphi(\mathbf{x}) = \frac{Q}{4\pi\epsilon_0} \ln \frac{1}{r} \quad (5.7)$$

berechnet. Die zugehörige Feldstärke wird durch die Beziehung:

$$\mathbf{E}(\mathbf{x}) = \frac{Q}{4\pi\epsilon_0} \frac{1}{r} \mathbf{e}_r \quad (5.8)$$

beschrieben. Der Vektor  $\mathbf{e}_r$  ist hierbei der Einheitsvektor in radialer Richtung:

$$\mathbf{e}_r = \frac{\mathbf{x}}{r}. \quad (5.9)$$

In Analogie zu den Beziehungen des elektrostatischen Feldes wird vereinbart, dass die Feldstärke  $\mathbf{E}$  für die nachfolgenden Ausführungen als Gradient  $\mathbf{G}$  mit der Beziehung:

$$\mathbf{G}(\mathbf{x}) = \frac{q}{r} \mathbf{e}_r \quad (5.10)$$

bezeichnet wird. Die Größe  $q$  ist ein Maß für die Stärke der Ladung. Der Wert für die negative Ladung im Zielpunkt wird zu  $-1$  gewählt. Die positive Ladung  $q$  des Objektes wird durch die Beziehung:

$$q = \frac{R}{R + D} \quad (5.11)$$

beschrieben. Dabei ist  $R$  der Radius des Hinderniskreises und  $D$  beschreibt den Abstand zwischen den beiden Ladungen. Durch Gleichung (5.11) wird garantiert, dass alle Gradientenlinien, welche innerhalb des Kreises beginnen, herausgeführt werden und im Zielpunkt enden. Gradientenlinien, die außerhalb des Kreises beginnen, verlaufen ausschließlich außerhalb des Kreises. Für die Beweisführung sei auf [71] und [72] verwiesen.

Für die Ermittlung des resultierenden Gradienten  $\mathbf{G}$  an der aktuellen Fahrzeugposition sind die beiden Gradienten für die positive und negative Ladung getrennt zu berechnen und durch eine Superposition zusammenzufassen:

- 1) Ermittlung des durch die negative Zielpunktladung erzeugten Gradienten  $\mathbf{G}_{\text{Ziel}}$  an der aktuellen Fahrzeugposition  $\mathbf{x}_{\text{Fahr}}$ .

$$\begin{aligned} \mathbf{x}_{\text{Ziel}}^{\text{Distanz}} &= \mathbf{x}_{\text{Ziel}} - \mathbf{x}_{\text{Fahr}}, \quad r_{\text{Ziel}} = |\mathbf{x}_{\text{Ziel}}^{\text{Distanz}}|, \quad \mathbf{e}_r^{\text{Ziel}} = \frac{\mathbf{x}_{\text{Ziel}}^{\text{Distanz}}}{r_{\text{Ziel}}} \\ \mathbf{G}_{\text{Ziel}} &= \frac{-1}{r_{\text{Ziel}}} \mathbf{e}_r^{\text{Ziel}} = \frac{-1}{r_{\text{Ziel}}^2} \mathbf{x}_{\text{Ziel}}^{\text{Distanz}} \end{aligned} \quad (5.12)$$

- 2) Ermittlung des durch die positive Hindernisladung erzeugten Gradienten  $\mathbf{G}_{\text{Hind}}$  an der aktuellen Fahrzeugposition  $\mathbf{x}_{\text{Fahr}}$ .

$$\begin{aligned} \mathbf{x}_{\text{Hind}}^{\text{Distanz}} &= \mathbf{x}_{\text{Fahr}} - \mathbf{x}_{\text{Hind}}, \quad r_{\text{Hind}} = |\mathbf{x}_{\text{Hind}}^{\text{Distanz}}|, \quad \mathbf{e}_r^{\text{Hind}} = \frac{\mathbf{x}_{\text{Hind}}^{\text{Distanz}}}{r_{\text{Hind}}} \\ \mathbf{G}_{\text{Hind}} &= \frac{R}{R + |\mathbf{x}_{\text{Ziel}} - \mathbf{x}_{\text{Hind}}|} \frac{1}{r_{\text{Hind}}} \mathbf{e}_r^{\text{Hind}} = \underbrace{\frac{R}{R + |\mathbf{x}_{\text{Ziel}} - \mathbf{x}_{\text{Hind}}|}}_q \frac{1}{r_{\text{Hind}}^2} \mathbf{x}_{\text{Hind}}^{\text{Distanz}} \end{aligned} \quad (5.13)$$

- 3) Berechnung des resultierenden Gradienten  $\mathbf{G}$  durch Superposition der einzelnen Gradienten.

$$\mathbf{G} = \mathbf{G}_{\text{Ziel}} + \mathbf{G}_{\text{Hind}}. \quad (5.14)$$

Abbildung 5.8 zeigt das so entstandene Potentialfeld mit den dazugehörigen Gradientenlinien. Da die Gradientenlinien das Hindernis tangieren, aber nicht schneiden können, muss der Radius  $R$  des Kreises um einen entsprechenden Betrag vergrößert werden, der eine Kollision zwischen Fahrzeug und Objekt verhindert. Hier bietet sich eine Vergrößerung mit der  $n$ -fachen maximalen Ausdehnung des Fahrzeuges an. Der so entstandene Kreis wird auch als Sicherheitszone bezeichnet [72].

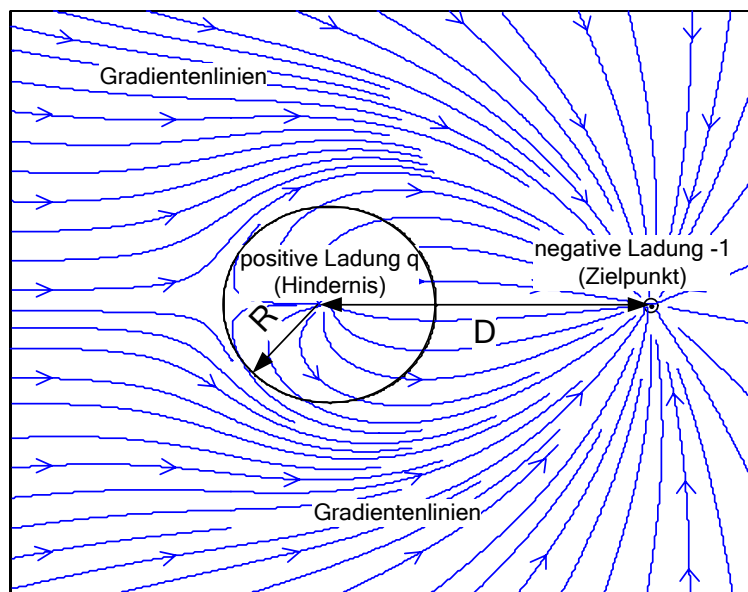


Abbildung 5.8: Gradientenlinien des Dipols

Befindet sich das Fahrzeug auf der dem Zielpunkt abgewandten Seite und fährt in der Nähe der Symmetrieachse des Dipols, wird es erst kurz vor Erreichen des Hinderniskreises zum Umfahren des Hindernisses gebracht (siehe Abbildung 5.8). Dabei sind starke Lenkkommandos notwendig, um das Fahrzeug auf der vorgegebenen Gradientenlinie zu halten. Des Weiteren entspricht die Form der Gradientenlinien in diesem Bereich und an den seitlichen Vorbeiführungen nicht einer wegoptimalen Fahrtroute. Diese Effekte nehmen mit größer werdenden Werten von  $R$  und  $D$  zu. Sie sollen deshalb durch die geometrische Konstruktion von Gradientenlinien zur wegoptimalen Führung in einem neu entwickelten Verfahren, das im nächsten Abschnitt detailliert beschrieben wird, beseitigt werden.

## 5.3 Verfahren der Geometrischen Konstruktion

Das in diesem Abschnitt beschriebene Verfahren basiert auf der in Abschnitt 5.2.4 im Überblick vorgestellten Arbeit [71, 72] von Guldner. Bei diesem Verfahren folgt das Fahrzeug den durch künstliche harmonische Dipol-Potentiale erzeugten Gradientenlinien. In den nächsten Abschnitten wird ein neu entwickeltes Verfahren vorgestellt, das die Gewinnung der Gradienten aus einer geometrischen Konstruktion der Gradientenlinien ermöglicht. Die anderen Funktionsprinzipien des Verfahrens der harmonischen Dipol-Potentiale werden übernommen und ggf. durch eigene Lösungsansätze modifiziert und erweitert. Ferner werden Möglichkeiten aufgezeigt, ein Ausweichen unter Einbeziehung der Geschwindigkeitsinformationen der Seeströmung und der Hindernisse durchzuführen.

### 5.3.1 Erzeugung von Gradientenlinien durch geometrische Konstruktion

Einführend sei an dieser Stelle erwähnt, dass die in diesem Abschnitt erzeugten Gradientenlinien keine physikalischen „Vorbilder“ besitzen. Sie sollen lediglich zu einer stetigen und wegoptimalen Fahrtroute führen. Hierzu wird das Operationsgebiet in einzelne Sektoren aufgeteilt (siehe Abbildung 5.9). Die Einteilung der Sektoren erfolgt in Abhängigkeit der Fahrzeugposition  $x_{Fahr}$  vom Hinderniszentrum  $x_{Hind}$  und vom Zielpunkt.

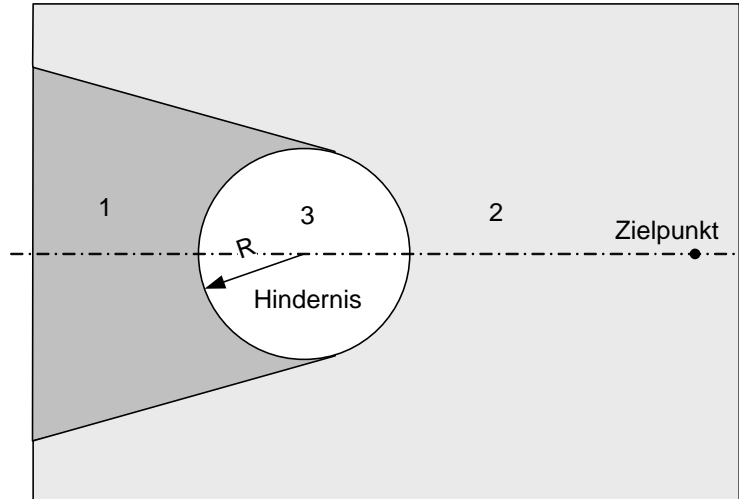


Abbildung 5.9: Einteilung des Operationsgebietes in Sektoren

Wenn die Bedingung ( $|\mathbf{x}_{Fahr} - \mathbf{x}_{Hind}| < R$ ) erfüllt ist, befindet sich das Fahrzeug im Inneren des Sicherheitskreises. Hier ist der Sektor 3 aktiv. Der Gradient wird dabei durch die Gleichung:

$$\mathbf{x}_{Hind}^{Distanz} = \mathbf{x}_{Fahr} - \mathbf{x}_{Hind}$$

$$\mathbf{G} = \frac{\mathbf{x}_{Hind}^{Distanz}}{|\mathbf{x}_{Hind}^{Distanz}|} \quad (5.15)$$

berechnet. Die Aktivierung des Sektors 3 scheint im ersten Moment ausgeschlossen. An dieser Stelle sei jedoch noch einmal festgestellt, dass der Radius R eine Sicherheitszone beschreibt, in der sich das Hindernis befindet. So ist es durchaus möglich, dass ein Fahrzeug in diesen Sicherheitskreis eindringen kann (starke Strömung, zu späte Erfassung des Hindernisses). In einem solchen Fall muss das Fahrzeug wieder sicher aus dem Kreis geführt werden, was eine Gradientenbeschreibung im Innern des Kreises erfordert.

Wird die Sicht zwischen Fahrzeug und Zielpunkt durch den Kreis verdeckt, ist keine direkte Anfahrt zum Zielpunkt möglich (Sektor 1). In diesem Fall muss der Richtungsvektor der Tangentengerade von  $\mathbf{x}_{Fahr}$  zum Kreis in Abhängigkeit der Lage von  $\mathbf{x}_{Fahr}$  zur Symmetrieachse bestimmt werden. Der Gradient ist dabei der normierte Richtungsvektor  $\mathbf{x}_{Tangente}^{Richtung}$ :

$$\mathbf{G} = \frac{\mathbf{x}_{Tangente}^{Richtung}}{|\mathbf{x}_{Tangente}^{Richtung}|} \quad (5.16)$$

Gibt es eine direkte Verbindung zwischen Fahrzeug und Zielpunkt, ist Sektor 2 aktiv. Dann gilt:

$$\mathbf{x}_{Ziel}^{Distanz} = \mathbf{x}_{Ziel} - \mathbf{x}_{Fahr}$$

$$\mathbf{G} = \frac{\mathbf{x}_{Ziel}^{Distanz}}{|\mathbf{x}_{Ziel}^{Distanz}|} \quad (5.17)$$

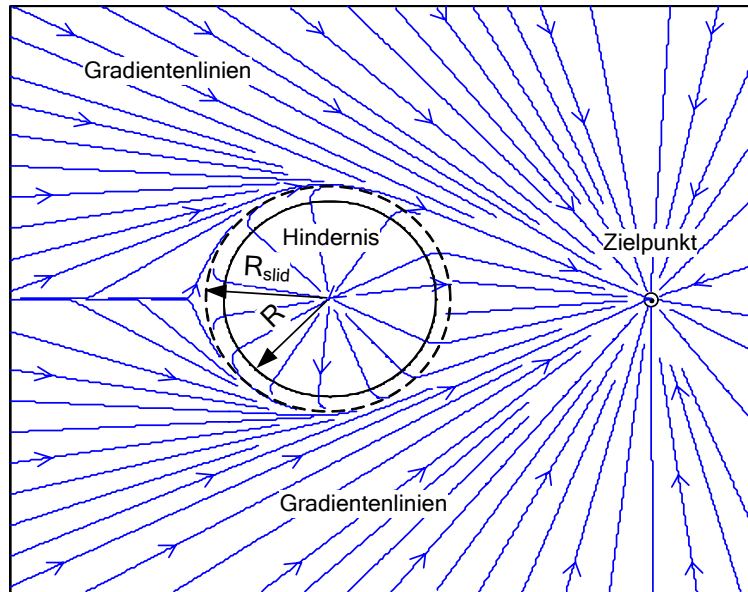


Abbildung 5.10: Gradientenlinien bei geometrischer Konstruktion

Um beim Austreten der Gradientenlinien aus dem Sicherheitskreis einen kontinuierlichen Übergang zu den Gradientenlinien der Sektoren 1 und 2 zu erreichen, wurde ein Radius  $R_{slid}$  eingeführt. In dem dadurch entstehenden Kreisring zwischen  $R$  und  $R_{slid}$  findet eine gewichtete Überlagerung unter Verwendung des Wichtungsfaktors  $\alpha$  zwischen den Gradienten außerhalb und innerhalb des Kreises nach der Beziehung:

$$\alpha = \frac{|\mathbf{x}_{Fahr} - \mathbf{x}_{Hind}| - R}{R_{slid} - R} \quad (5.18)$$

$$\mathbf{G} = \alpha \underbrace{\mathbf{G}_{Sektor1-2}}_{\text{Außerhalb des Kreises}} + (1 - \alpha) \underbrace{\mathbf{G}_{Sektor3}}_{\text{Innerhalb des Kreises}}$$

statt. Abbildung 5.10 zeigt ein so erzeugtes Gradientenfeld.

### 5.3.2 Einbeziehung der Geschwindigkeitsinformationen

Unter realen Umweltbedingungen kann es vorkommen, dass ein Hindernis sich bewegt (beschrieben durch  $\mathbf{v}_{Hind}$ ) oder eine starke Seeströmung  $\mathbf{v}_{Strömung}$  vorhanden ist. In solchen Fällen ist es sinnvoll, diese Geschwindigkeitsinformationen bei der Generierung der Gradientenlinien mit einfließen zu lassen. So soll es möglich sein, ein Ausweichmanöver schon früher zu beginnen, wenn sich ein Hindernis auf das Fahrzeug zubewegt oder eine starke Strömung in Fahrtrichtung vorhanden ist. In beiden Fällen wird ein mögliches Kollidieren durch unzureichendes Reagieren vermieden. Eine Möglichkeit, dies zu erreichen, ist die Vergrößerung der Sicherheitszone in Richtung einer resultierenden Geschwindigkeit  $\mathbf{v}_{res}$ , die durch die folgende Beziehung beschrieben wird:

$$\mathbf{v}_{res} = \mathbf{v}_{Hind} - \mathbf{v}_{Strömung} \quad (5.19)$$

Die Sicherheitszone wird damit von ihrer Kreisform mit dem Radius  $R$  in eine Ellipse überführt (siehe Abbildung 5.11).



Die Berechnung der Gradientenlinien erfolgt dann, wie in Abschnitt 5.3.1 beschrieben, für kreisförmige Hindernisse im Bildbereich mit anschließender Rücktransformation in den Originalbereich. Die Transformation für einen beliebigen Punkt  $\mathbf{x}$  in diesem Bildbereich setzt sich so aus mehreren Teilschritten zusammen (siehe Abbildung 5.12):

- 1) Translation des Punktes um die Position  $\mathbf{x}_{\text{Ellipse}}$  (Verschiebung der Ellipse in den Koordinatenursprung des Bildbereiches):

$$\mathbf{x}_{\text{trans}} = \mathbf{x} - \mathbf{x}_{\text{Ellipse}} \quad (5.23)$$

- 2) Rotation des Punktes um den Rotationswinkel der Ellipse  $\theta_{\text{Ellipse}}$  (Rotation der Ellipse zum Erreichen einer achsenparallelen Lage der Ellipse im Bildbereich):

$$\mathbf{x}_{\text{trans\_rot}} = \begin{bmatrix} \cos \theta_{\text{Ellipse}} & \sin \theta_{\text{Ellipse}} \\ -\sin \theta_{\text{Ellipse}} & \cos \theta_{\text{Ellipse}} \end{bmatrix} \mathbf{x}_{\text{trans}} \quad (5.24)$$

- 3) Skalierung des Punktes mit den inversen Werten der halben Haupt- und Nebenachse ( $a_{\text{Ellipse}}$  und  $b_{\text{Ellipse}}$ ) der Ellipse (Erreichen einer Kreisform für die Ellipse mit  $R = 1$ ):

$$\tilde{\mathbf{x}} = \begin{bmatrix} \frac{1}{a_{\text{Ellipse}}} & 0 \\ 0 & \frac{1}{b_{\text{Ellipse}}} \end{bmatrix} \mathbf{x}_{\text{trans\_rot}} \quad (5.25)$$

Nach Durchführung dieser Transformation können die notwendigen Berechnungen im Bildbereich durchgeführt werden. Der hierbei berechnete Gradient  $\mathbf{G}$  muss anschließend wieder vom Bildbereich in den Originalbereich rücktransformiert werden [Gleichung (5.26) und (5.27)].

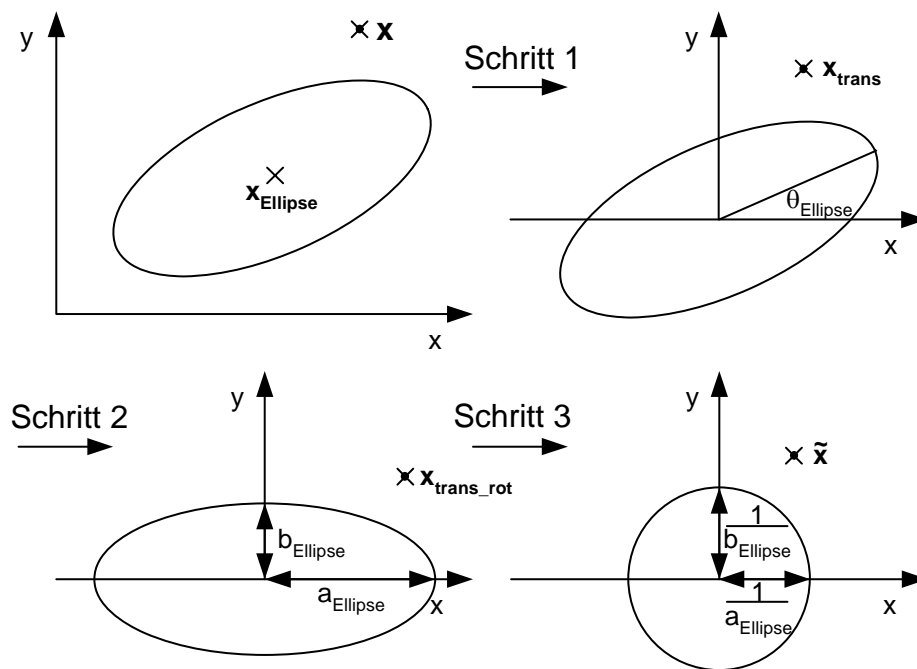


Abbildung 5.12: Schritte der Transformation

Dazu findet eine Rückskalierung nach Gleichung (5.26) mit anschließender Rückrotation nach Gleichung (5.27) statt. Es gilt:

$$\mathbf{G}_{rescal} = \begin{bmatrix} a_{Ellipse} & 0 \\ 0 & b_{Ellipse} \end{bmatrix} \tilde{\mathbf{G}} \quad (5.26)$$

$$\mathbf{G} = \begin{bmatrix} \cos \theta_{Ellipse} & -\sin \theta_{Ellipse} \\ \sin \theta_{Ellipse} & \cos \theta_{Ellipse} \end{bmatrix} \mathbf{G}_{rescal} \cdot \quad (5.27)$$

Bei den bisherigen Ausführungen wurde nur ein einzelnes Hindernis im Operationsgebiet betrachtet. Die nächsten Abschnitte beschreiben das Verfahren mit mehreren Hindernissen.

### 5.3.4 Anwendung für mehrere Hindernisse

In Analogie zum Verfahren der Harmonischen Dipol-Potentiale [72] werden zur Bestimmung des resultierenden Gradienten bei mehr als einem Hindernis nur die Gradientenlinien des zum Fahrzeug nächstgelegenen Hindernisses verwendet. Dies führt dazu, dass bei der Durchfah- rung eines Hindernisparcours ein Umschalten zum jeweils aktuellen Gradientenlinienfeld erfolgt. Um beim Übergang zwischen zwei Gradientenfeldern einen homogenen Verlauf zu erzielen, ist es notwendig, statt des binären Umschaltens ein „kontinuierliches“ Umschalten durchzuführen. In [72] wurde ein Bereich entlang der Grenzlinie zweier benachbarter Gradientenlinienfelder definiert ( $d_{Linie\_max}$ ), in dem sich die Gradienten der einzelnen Felder mittels eines Wichtungsfaktors  $\alpha$  linear überlagern. Dieser Wichtungsfaktor wurde proportional zum Fahrzeugabstand von der Grenzlinie gewählt. Eine vereinfachte Bestimmung dieses Abstandes kann durch Gleichung (5.28) auf Grundlage von Abbildung 5.13 erfolgen.

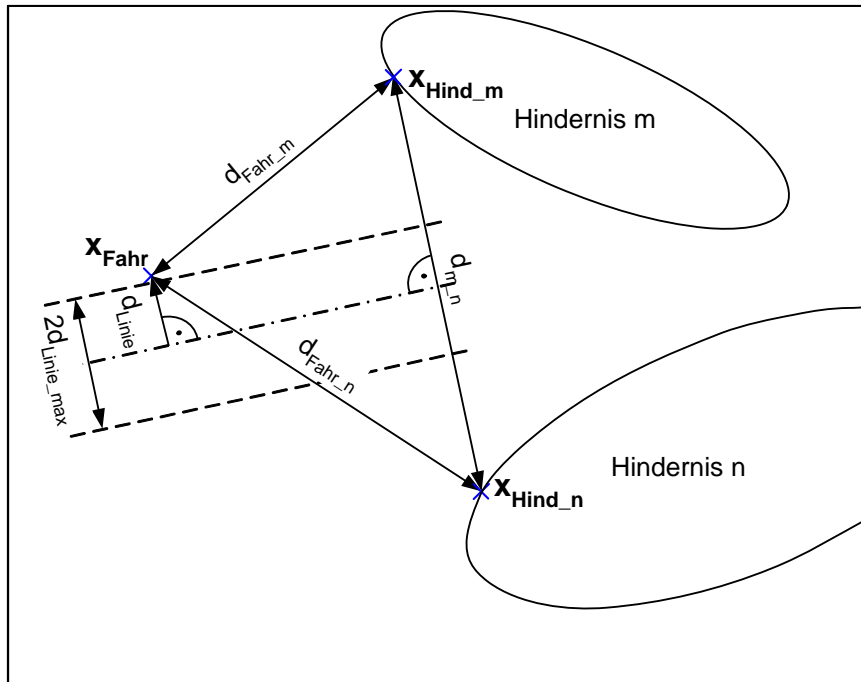


Abbildung 5.13: Vereinfachte Bestimmung von  $d_{Linie}$



Es gilt:

$$d_{Linie} = \frac{d_{Fahr\_n}^2 - d_{Fahr\_m}^2}{2d_{m\_n}}. \quad (5.28)$$

Die Abstände  $d_{Fahr\_n}$  und  $d_{Fahr\_m}$  beschreiben die kürzeste Entfernung von der aktuellen Fahrzeugposition zum jeweiligen Hindernis. Der resultierende Gradient bestimmt sich so durch die folgenden Beziehungen:

$$\mathbf{G} = \begin{cases} \mathbf{G}_{Hind\_m} & \text{für } d_{Linie} > d_{Linie\_max} \\ \mathbf{G}_{Hind\_n} & \text{für } d_{Linie} < -d_{Linie\_max} \\ \alpha \mathbf{G}_{Hind\_m} + (1-\alpha) \mathbf{G}_{Hind\_n} & \text{für } |d_{Linie}| < d_{Linie\_max} \end{cases} \quad (5.29)$$

mit dem Wichtungsfaktor  $\alpha$ :

$$\alpha = 0.5 \left( 1 + \frac{d_{Linie}}{d_{Linie\_max}} \right). \quad (5.30)$$

Sind mehr als zwei Hindernisse im Arbeitsraum vorhanden, wird aus den drei zum Fahrzeug nächstgelegenen Hindernissen im Bereich, in dem sich alle drei Übergangsbereiche überlappen (grau gekennzeichnet in Abbildung 5.14), der resultierende Gradient  $\mathbf{G}$  aus der gewichteten Überlagerung der drei Gradientenlinienfelder nach den Gleichungen (5.32)-(5.34) bestimmt. Ein so entstandenes Gradientenlinienfeld ist in Abbildung 5.15 dargestellt. Die Aktivierungsbedingung zur Durchführung dieser Berechnungen lautet:

$$\left( (|d_{Linie\_12}| < d_{Linie\_max}) \wedge (|d_{Linie\_13}| < d_{Linie\_max}) \wedge (|d_{Linie\_23}| < d_{Linie\_max}) \right). \quad (5.31)$$

$$\begin{aligned} \alpha_{12} &= 0.5 \left( 1 + \frac{d_{Line\_12}}{d_{Linie\_max}} \right) \\ \alpha_{23} &= 0.5 \left( 1 + \frac{d_{Line\_23}}{d_{Linie\_max}} \right) \\ \alpha_{13} &= 0.5 \left( 1 + \frac{d_{Line\_13}}{d_{Linie\_max}} \right) \end{aligned} \quad (5.32)$$

$$\begin{aligned} \beta_1 &= \frac{2}{3} \alpha_{13} + \frac{1}{3} \alpha_{12} \alpha_{23} - \frac{1}{3} \alpha_{13} \alpha_{23} \\ \beta_2 &= \frac{2}{3} \alpha_{23} + \frac{1}{3} \alpha_{13} + \frac{1}{3} \alpha_{12} \alpha_{13} - \frac{1}{3} \alpha_{13} \alpha_{23} \end{aligned} \quad (5.33)$$

$$\begin{aligned} \beta_3 &= 1 - \alpha_{13} - \frac{2}{3} \alpha_{23} + \frac{2}{3} \alpha_{13} \alpha_{23} - \frac{1}{3} \alpha_{12} \alpha_{23} + \frac{1}{3} \alpha_{12} \alpha_{13} \\ \mathbf{G} &= \beta_1 \mathbf{G}_1 + \beta_2 \mathbf{G}_2 + \beta_3 \mathbf{G}_3 \end{aligned} \quad (5.34)$$

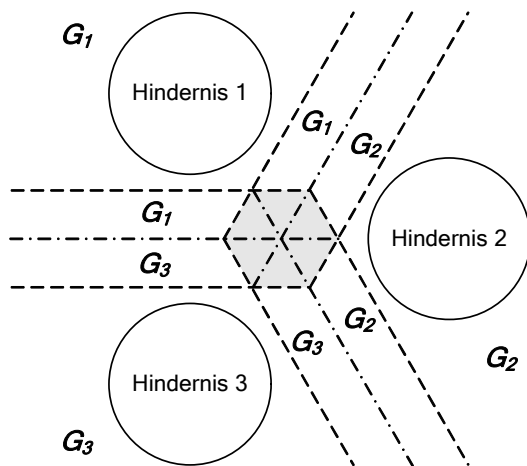


Abbildung 5.14: Überlagerung von drei Gradientenlinienfeldern

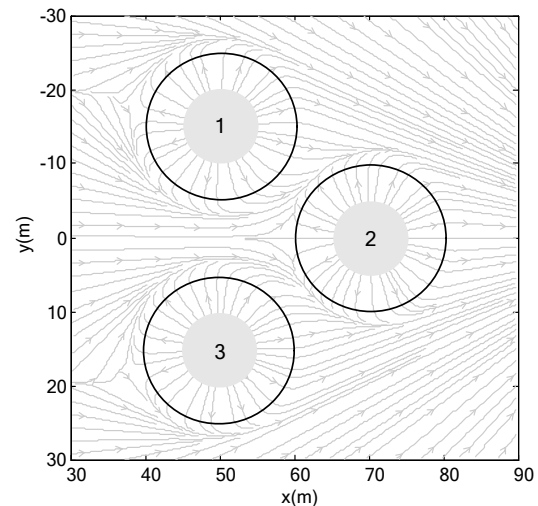


Abbildung 5.15: Gradientenlinienfeld

Ist die Bedingung (5.31) nicht aktiv, wird der Gradient nach Gleichung (5.29) bestimmt. Wird diese o. g. Berechnung des Gradienten nicht durchgeführt und statt dessen der Gradient nur aus den Gradientenlinienfeldern der zwei nächstgelegenen Hindernisse bestimmt, kommt es zu Unstetigkeiten des Kurskommandos  $\psi$  beim Hineinfahren in einen anderen Übergangsbereich. Abbildung 5.16 zeigt dies am Vergleich bei der Verwendung von zwei und drei Gradientenlinienfeldern. Als Hindernisszenario wurde das in Abbildung 5.15 Dargestellte verwendet. Befindet sich das Fahrzeug im vorgegebenen Hindernisszenario zwischen dem Hindernis 1 und Hindernis 3, kann bei Verwendung von drei Gradientenlinienfeldern schon früher mit dem Umfahren des Hindernisses 2 begonnen werden. Das Fahrzeug wird sozusagen nicht mehr so lange durch den gebildeten Gradienten des Übergangsbereiches zwischen Hindernis 1 und 3 zwangsgeführt.

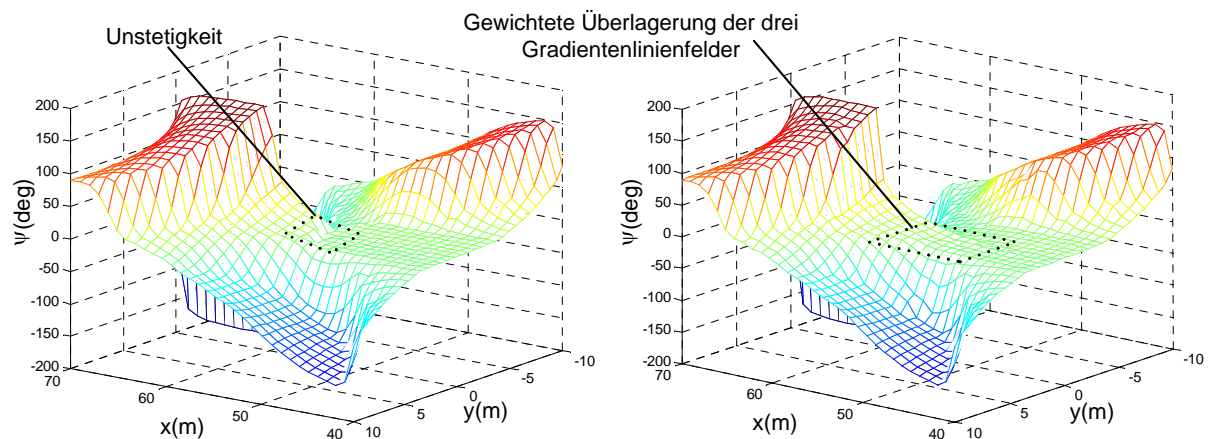


Abbildung 5.16: Resultierender Kurs bei Verwendung einer gewichteten Überlagerung von zwei (linke Abbildung) und drei (rechte Abbildung) Gradientenlinienfeldern

### 5.3.5 Überlappende Sicherheitsbereiche

Sind mehrere Hindernisse im Operationsgebiet vorhanden, muss darauf geachtet werden, dass sich die Sicherheitsbereiche der einzelnen Hindernisse nicht überlappen. Lässt sich dies nicht vermeiden, müssen die entsprechenden Hindernisse durch eine gemeinsame Sicherheitszone abgedeckt werden [72]. Findet eine solche Zusammenfassung nicht statt, könnte das Fahrzeug durch die sich überlappenden Sicherheitsbereiche hindurch fahren, da es stets durch die Gradientenlinien des nächstgelegenen Hindernisses geführt wird. Ein Lösungsansatz zur Zusammenfassung von einzelnen Sicherheitsbereichen wird in den folgenden Abschnitten vorgestellt. Die Vorgaben an eine solche Hüllsicherheitszone sind die, dass sie die einzelnen sich überlappenden Sicherheitsbereiche vollkommen einschließt und dabei möglichst wenig Fläche einnimmt. Des Weiteren muss dieser Sicherheitsbereich eine Kreis- bzw. Ellipsenform besitzen, da mit dieser Form die Bildung des Gradientenlinienfeldes durchgeführt wird. Bei den ersten Überlegungen wurde nach einer analytischen Möglichkeit gesucht, dieses o. g. Problem für zwei sich überlappende Ellipsen zu lösen. Eigene Untersuchungen und Recherchen sowie die Korrespondenz mit Eberly [43] und Schönherr [136] zeigten, dass keine analytische Methode bekannt ist, eine solche Ellipse zu berechnen. Zur Lösung dieses Problems wird der Ansatz zur „Berechnung kleinster Ellipsoide um Punktmengen“ verwendet [62]. Nachfolgend werden die einzelnen Schritte zur Bestimmung der Sicherheitszonen vorgestellt.

#### 5.3.5.1 Bildung der Hüllsicherheitszonen

Die Bildung der Hüllsicherheitszonen besteht aus einem mehrstufigen, iterativen Prozess. In einer ersten Stufe werden die Objektnummern der sich überlappenden Sicherheitszonen in einem Objektcluster zusammengefasst. So entsteht ein Objektcluster-Vektor, dessen Elementanzahl der der zu bildenden Hüllsicherheitszonen entspricht. Die einzelnen Elemente des Objektcluster-Vektors bestehen aus Vektoren, welche die jeweiligen zu umhüllenden Sicherheitszonen beinhalten. Abbildung 5.17 zeigt die Bildung des Objektcluster-Vektors. Dazu findet eine paarweise Überlappungsüberprüfung aller Sicherheitsbereiche im Operationsgebiet statt. Diese Überprüfung verwendet das im Abschnitt 2.3.3 vorgestellte Zwei-Phasen-Konzept. In der ersten Phase findet ein einfacher Überlappungstest mit den achsengerichteten Boxen der Sicherheitszonen statt. Erst wenn dieser Test der überprüften Sicherheitszone erfolgreich war, werden ihre Ellipsen auf Überschneidung überprüft. Die Herleitung befindet sich im Abschnitt 5.3.5.2. Findet eine Überschneidung zwischen zwei Sicherheitszonen statt, wird überprüft, ob ihre Objektnummern schon in einem der gebildeten Cluster enthalten sind. Sind beide noch nicht enthalten, wird ein neues Clusterelement angelegt. Ist nur eine Objektnummer in einem der Cluster enthalten, wird die andere Nummer diesem Cluster hinzugefügt. Sind beide Objektnummern in unterschiedlichen Clustern enthalten, werden die beiden Cluster zusammengefasst. Nach der Erzeugung des Objektcluster-Vektors werden die Hüllsicherheitszonen der einzelnen Clusterelemente des Vektors nach Abschnitt 5.3.5.3 gebildet. Wurde während der o. g. Überprüfung kein Clusterelement angelegt, d.h. der Objektcluster-Vektor ist leer, kann die Überprüfung abgebrochen werden. Anderenfalls wird mit den gebildeten Hüllsicherheitszonen und den noch keinem Cluster angehörigen Sicherheitszonen eine weitere Überprüfung durchgeführt. Dies ist notwendig, da die gebildeten Hüllsicherheitszonen eine andere Sicherheitszone überlappen können. Die Überprüfung wird solange wiederholt bis der Objektcluster-Vektor leer ist.



mit den Parametern:

$$\begin{aligned}
 h_0 &= d_{00}d_{10} - d_{20}^2 \\
 h_1 &= d_{01}d_{10} + d_{00}d_{11} - 2d_{20}d_{21} \\
 h_2 &= d_{01}d_{11} + d_{00}d_{12} - d_{21}^2 - 2d_{20}d_{22} \\
 h_3 &= d_{01}d_{12} + d_{00}d_{13} - 2d_{21}d_{22} \\
 h_4 &= d_{01}d_{13} - d_{22}^2
 \end{aligned} \tag{5.40}$$

Die einzelnen Variablen  $d_{xx}$  werden durch einen Koeffizientenvergleich der Gleichungen (5.36) und (5.37) mit den im Anhang beschriebenen, vollständig besetzten, quadratischen Gleichungen zu:

$$\begin{aligned}
 d_{00} &= a_{\text{EllipseA}}d_{\text{EllipseB}} \\
 d_{01} &= a_{\text{EllipseA}}b_{\text{EllipseB}} - a_{\text{EllipseB}}b_{\text{EllipseA}} \\
 d_{10} &= -d_{\text{EllipseB}}f_{\text{EllipseA\_center}} \\
 d_{11} &= b_{\text{EllipseA}}f_{\text{EllipseB}} - b_{\text{EllipseB}}f_{\text{EllipseA}} \\
 d_{12} &= b_{\text{EllipseA}}e_{\text{EllipseB}} - d_{\text{EllipseB}}c_{\text{EllipseA}} \\
 d_{13} &= b_{\text{EllipseA}}c_{\text{EllipseB}} - b_{\text{EllipseB}}c_{\text{EllipseA}} \\
 d_{20} &= a_{\text{EllipseA}}f_{\text{EllipseB}} - a_{\text{EllipseB}}f_{\text{EllipseA\_center}} \\
 d_{21} &= a_{\text{EllipseA}}e_{\text{EllipseB}} \\
 d_{22} &= a_{\text{EllipseA}}c_{\text{EllipseB}} - a_{\text{EllipseB}}c_{\text{EllipseA}}
 \end{aligned} \tag{5.41}$$

bestimmt. Für die weiterführenden Schritte sei auf Abschnitt 2.3.2.2 verwiesen.

### 5.3.5.3 Erzeugung der Ellipse einer Hülsicherheitszone

Kommt es zu Überschneidungen zwischen zwei oder mehreren Sicherheitszonen, müssen diese Gebiete durch eine gemeinsame Sicherheitszone umhüllt werden. Diese Zone sollte eine Ellipsenform besitzen, welche die einzelnen Sicherheitszonen vollständig einschließt, jedoch möglichst wenig befahrbare Gebiet enthält, also eine minimale Fläche besitzt. Da, wie im einführenden Teil des Abschnittes 5.3.5 schon erläutert, keine analytische Lösung bekannt ist, wurde zur Lösung dieses Problems der Ansatz zur „Berechnung kleinster Ellipsoide um Punktmengen“ verwendet [62]. Dazu wird um jede Sicherheitszone ein Polygon definiert, welches die Sicherheitszone einschließt. Um die Stützstellen aller so gebildeten Polygone wird eine Ellipse unter Verwendung der CGAL Bibliothek [34] berechnet. Diese Ellipse zeichnet sich dadurch aus, dass sie alle Stützstellen enthält und dabei die kleinste mögliche Fläche einnimmt. Durch diesen Ansatz können gleich mehrere, sich überlappende Sicherheitszonen zu einer Ellipse zusammengefasst werden.

Die Berechnung der Polygonstützstellen erfolgte auf der Basis von normierten Stützstellen um einen Einheitskreis. Diese werden durch Skalierung und Rotation der entsprechenden Ellipsenform angepasst (siehe Gleichung (5.26) und (5.27)). Die Stützstellen  $S_{0...n-1}$  werden für den ersten Quadranten des Einheitskreises durch Schnittpunktberechnung nach Gleichung (5.43) unter Verwendung der in Gleichung (5.42) definierten Strahlen berechnet und für die anderen Quadranten durch entsprechenden Vorzeichenwechsel angepasst (siehe Abbildung 5.18).

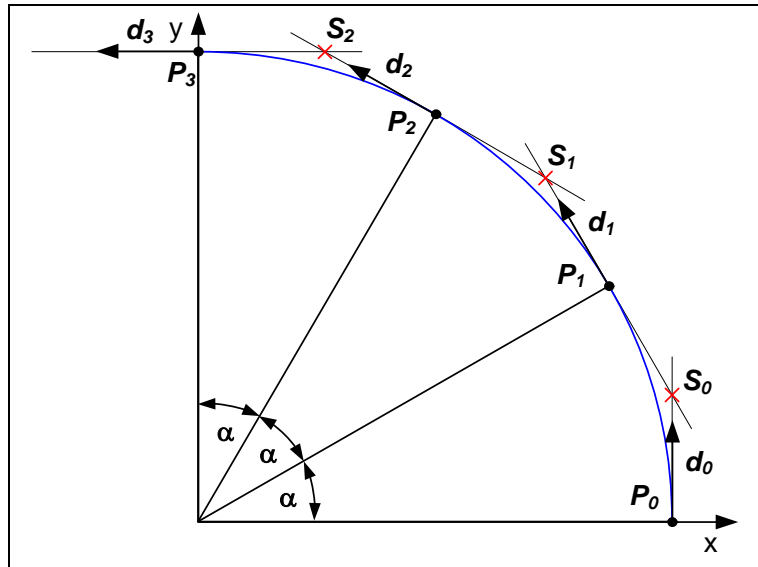


Abbildung 5.18: Ermittlung der Stützstellen auf einen Einheitskreis

Somit gilt:

$$P_n = \begin{bmatrix} \cos(n \cdot \alpha) \\ \sin(n \cdot \alpha) \end{bmatrix} \quad d_n = \begin{bmatrix} -\sin(n \cdot \alpha) \\ \cos(n \cdot \alpha) \end{bmatrix} \quad (5.42)$$

$$E = P_{n+1} - P_n \quad t = \frac{E \times d_{n+1}}{d_n \times d_{n+1}} \quad (5.43)$$

$$S_n = P_n + t \cdot d_n$$

Abbildung 5.19 zeigt die ermittelten Stützstellen der einzelnen Sicherheitszonen und die daraus mit der CGAL-Bibliothek berechnete Ellipse. In der Vergrößerung ist deutlich zu erkennen, dass die Polygonkanten die Sicherheitszone tangieren aber nicht schneiden.

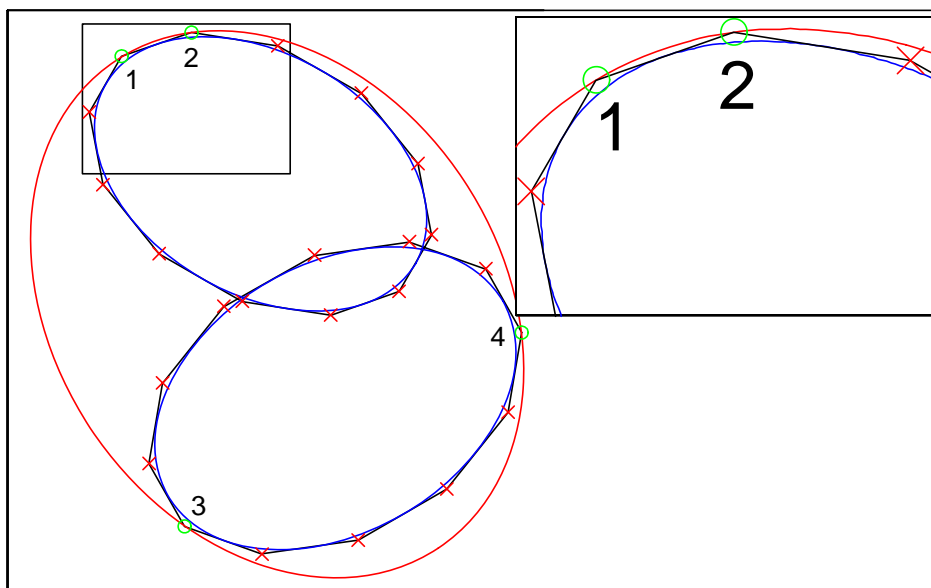


Abbildung 5.19: Kleinste eingeschlossene Ellipse

### 5.3.5.4 Bildung des Gradienten im Inneren der Hüllsicherheitszone

Befindet sich das Fahrzeug im Inneren der Hüllsicherheitszone, wird der Gradient so bestimmt, als ob das Fahrzeug im Inneren der Sicherheitszone liegen würde. Das heißt, der Gradient ist vom Mittelpunkt der Sicherheitszone weggerichtet. Einen anzufahrenden Zielpunkt gibt es im Inneren der Hüllsicherheitszone nicht. Die Berechnungen im Grenzbereich zwischen zwei benachbarten Gradientenlinienfeldern erfolgen nach den Beziehungen im Abschnitt 5.3.4. Durch diese Vorschriften wird das Fahrzeug sicher aus der Hüllsicherheitszone geführt. Abbildung 5.20 zeigt das so gebildete Gradientenfeld.

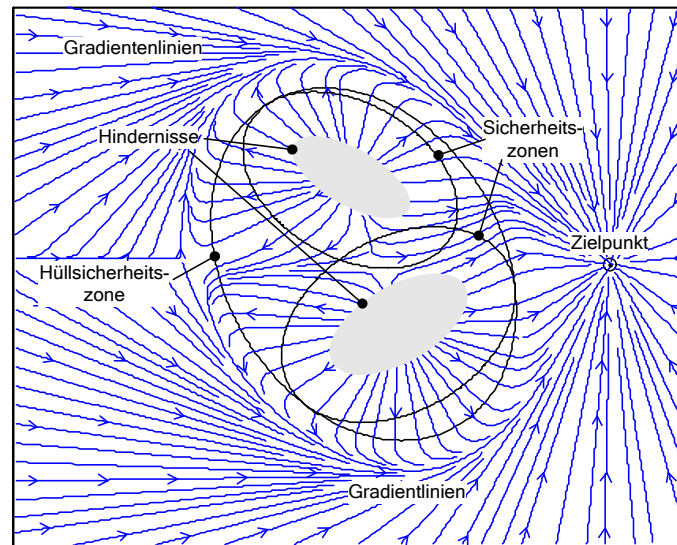


Abbildung 5.20: Gradientenlinien bei zusammengefassten Sicherheitszonen

## 5.4 Test des Verfahrens

### 5.4.1 Vergleich der Verfahren

In diesem Abschnitt wird das neu entwickelte Verfahren der Geometrischen Konstruktion mit dem Verfahren der Harmonischen Dipol-Potentiale verglichen. Dazu wurde ein Hindernisparcours definiert, der in Abbildung 5.22 dargestellt ist. Die abzufahrenden Solltrajektorien, welche den Gradientenlinien entsprechen, wurden durch Integration des Gradienten bestimmt. Das Fahrzeug soll mit einer Marschgeschwindigkeit von 2 m/s diesen Parcours durchfahren. Es zeigte sich, dass bei Verwendung des neu entwickelten geometrischen Verfahrens die Sollkursrate  $r$  (Drehgeschwindigkeit des Sollkurses) wesentlich geringere Maximalwerte als bei dem Verfahren der Harmonischen Dipol-Potentiale besitzt (siehe Abbildung 5.21). Dies ist für den praktischen Einsatz ein wichtiges Kriterium, da das eingesetzte Fahrzeug schnellen Sollkursänderungen nicht folgen kann, was zum Abweichen von der Gradientenlinie und damit zu einer Kollision führen kann. Des Weiteren bewirkt jede schnelle Sollkursänderung eine Vergrößerung des Navigationsfehlers. Schon aus diesem Grund sollten derartige Manöver vermieden werden. Die benötigte Weglänge zum Durchfahren des Parcours beträgt beim Geometrischen Verfahren bis zu 9 % weniger als bei der Methode der harmonischen Dipol-Potentiale. Die Solltrajektorienverläufe des geometrischen Verfahrens sind der Form von optimalen Wegen, wie sie in [152] unter Verwendung einer Evolutionsstrategie generiert wurden, sehr ähnlich.

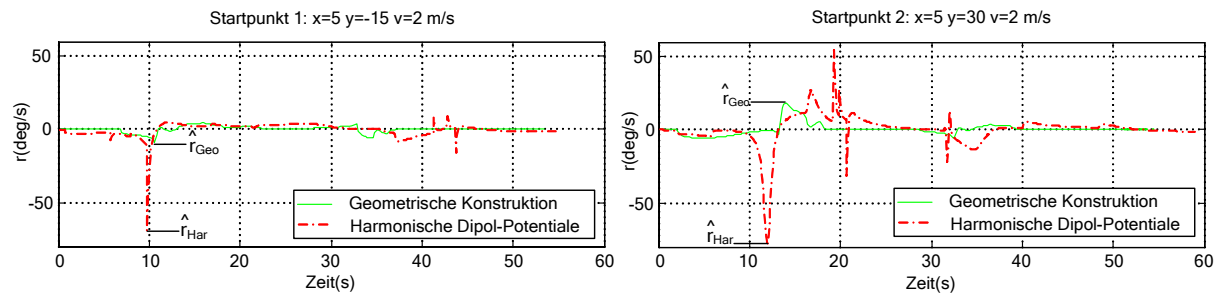


Abbildung 5.21: Auswertung der Sollkursraten

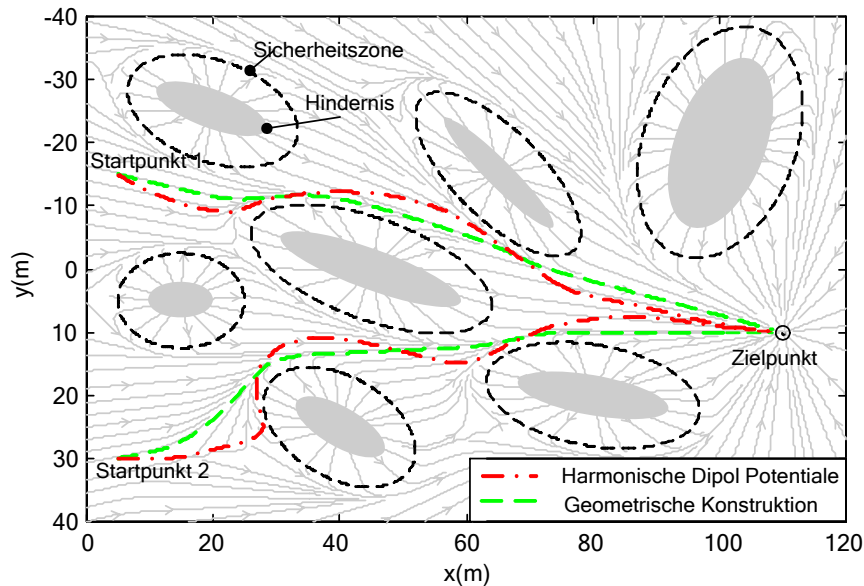


Abbildung 5.22: Wege durch einen Hindernisparcours

## 5.4.2 Führung des Fahrzeuges

Ein Folgen der im Abschnitt 5.4.1 beschriebenen Solltrajektorien bzw. Gradientenlinien erfordert nach [71] die exakte Ausrichtung des Geschwindigkeitsvektors des Fahrzeuges am Gradientenvektor. Dazu wird in [71] ein Gleitregler zum Gradientenfolgen eingesetzt. In dieser Arbeit erfolgt die Führung der berechneten Gradientenrichtung durch die Verwendung einer Kursregelung. In diesem Abschnitt soll die Notwendigkeit einer solchen Kursregelung im Vergleich zu einer Gierwinkelregelung erläutert werden. Dazu werden zuerst Definitionen für die in dieser Arbeit verwendeten Begriffe Kurs und Gierwinkel gegeben.

**Definition 1 (Kurs):** Unter Kurs versteht man den Winkel zwischen der Bewegungsrichtung eines Schiffs oder Flugzeugs und einer Bezugsrichtung [155]. Nach [118] lautet die exakte Benennung des in dieser Arbeit verwendeten Kursbegriffes *Kurs über Grund* (engl.: *course over the ground*). Die Definition dafür lautet dort: Richtung der während der Fahrt gemessenen Fahrzeugbewegung bezüglich der Erdoberfläche. Winkel von rechtweisend Nord bis zur Richtung des Weges über Grund [118].

**Definition 2 (Gierwinkel):** Der Gierwinkel beschreibt die Ausrichtung des Fahrzeuges in der Horizontalebene. Die Bezeichnung für diesen Winkel lautet nach [118] *rechtweisender Kurs* (engl.: *course to steer, true heading*). Andere Bezeichnungen sind Azimut und Steuerkurs. In [118] ist dieser Winkel wie folgt definiert: Winkel von rechtweisend Nord (True North) bis zur Rechtvorausrichtung des Fahrzeuges.



Zur Überprüfung des Verhaltens einer Kurs- und Gierwinkelregelung werden Simulationen unter Verwendung der im Kapitel 9 beschriebenen Simulationsumgebung mit dem AUV M62 (siehe Abschnitt 1.1.2) als Versuchsträger durchgeführt. Dabei wird das in Abbildung 5.23 dargestellte Hindernis ohne und mit vorhandener Seeströmung von 1,5 m/s umfahren. Die Fahrzeuggeschwindigkeit durchs Wasser beträgt 2 m/s. Die bei Verwendung der Kursregelung ohne vorhandene Seeströmung entstandene Trajektorie entspricht dem gewünschten Verhalten bzw. den Gradientenlinien des Verfahrens der Geometrischen Konstruktion. Die in einigen Arbeiten verwendete Gierwinkelregelung zur Führung des Fahrzeuges kann in strömungsfreien Gebieten als Kompromiss zur Kursregelung eingesetzt werden. Bei einer starken Strömung ist ihr Einsatz jedoch nicht möglich.

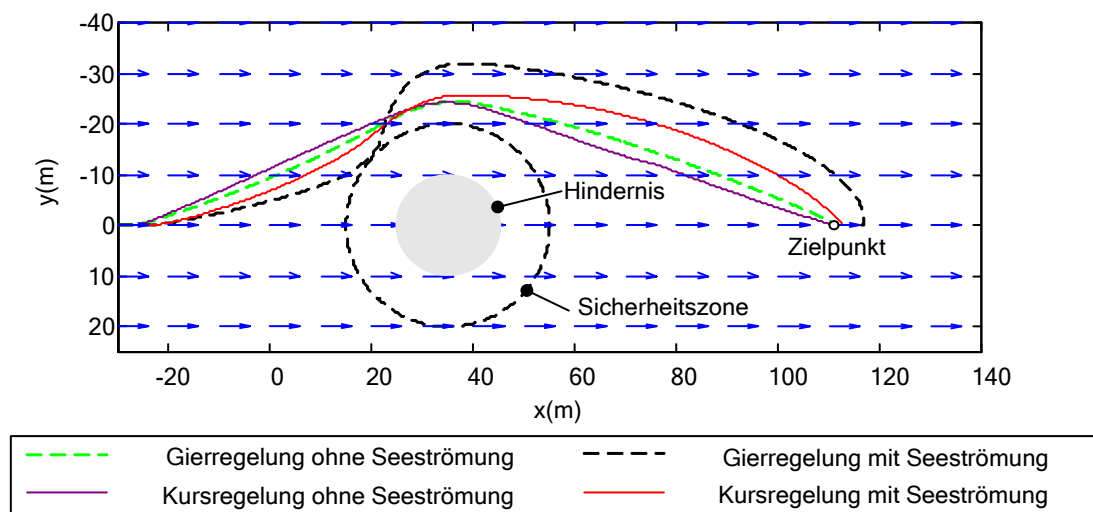


Abbildung 5.23: Ausweichtrajektorien bei Verwendung einer Kurs- und Gierwinkelregelung

### 5.4.3 Laufzeitmessungen

In diesem Abschnitt sollen einige Laufzeituntersuchungen des neu entwickelten Verfahrens der Geometrischen Konstruktion vorgestellt werden. Die Messbedingungen entsprechen denen des Abschnittes 2.4. Für die Tests wird der in Abbildung 5.24 abgebildete Hindernisparcours verwendet. Dabei werden einige Sicherheitszonen so platziert, dass sie sich überlappen. Dies dient zur Laufzeitüberprüfung der Algorithmen zur Zusammenfassung von überlappenden Sicherheitszonen (siehe Abschnitt 5.3.5). Die Testszenarien sind in Tabelle 5.1 enthalten.

Tabelle 5.1: Testszenarien

Test	Anzahl der Objekte
Test 1	10
Test 2	20
Test 3	30
Test 4	40
Test 5	50
Test 6	60
Test 7	70
Test 8	80
Test 9	90
Test 10	100

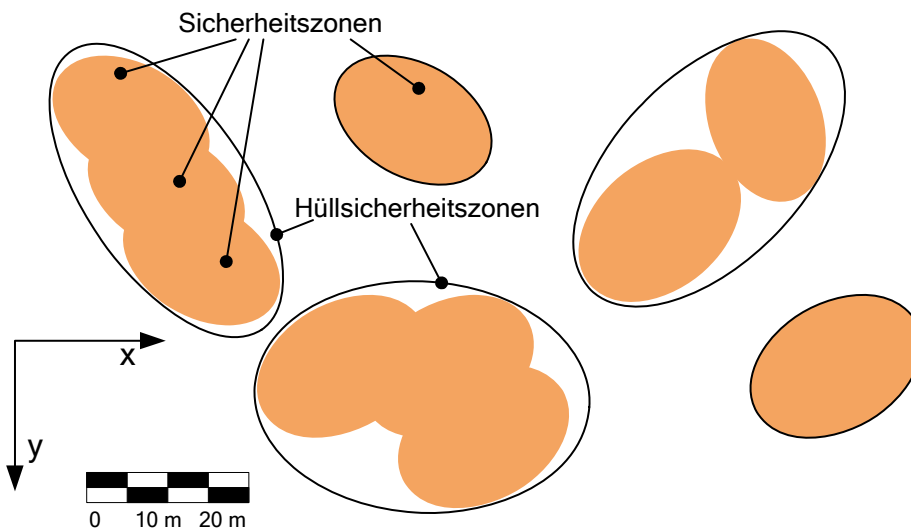


Abbildung 5.24: Hindernisparcours

In Abbildung 5.24 sind die mit diesem Verfahren erzeugten Hüllsicherheitszonen dargestellt. Die horizontale Anordnung der Sicherheitszonen der Tests 1 bis 10 entspricht der Abbildung 5.24. In der Vertikalen findet eine kaskadenförmige Anordnung der einzelnen Zonen statt. Hier wurden immer 10 Sicherheitszonen in einer Ebene platziert.

Das Diagramm in Abbildung 5.25 zeigt die Rechenzeiten zur Ermittlung der Hüllsicherheitszonen unter Verwendung des Verfahrens der kleinsten eingeschlossenen Ellipse [34, 62], des Verfahrens der kleinsten eingeschlossenen Kreise [34, 51] und eines Verfahrens zur Bildung eines Hüllkreises aus den achsengerichteten Begrenzungsboxen (AABB→Kreis). Das Verfahren der kleinsten eingeschlossenen Ellipse ist das Rechenaufwendigste. Durch seine Verwendung ist es jedoch möglich, die einzelnen überlappenden Sicherheitsbereiche so zusammenzufassen, dass möglichst wenig befahrbare Bereiche mit eingeschlossen werden. Bei Verwendung der anderen beiden Verfahren entstand für den in Abbildung 5.24 dargestellten Hindernisparcours eine einzige, alle Sicherheitszonen umschließende Hüllsicherheitszone. Für die in dieser Arbeit vorgegebene maximal mögliche Objektanzahl von 50 (Test 5) beträgt die Rechenzeit 12,42 ms. Diese Berechnungen müssen jedoch nur bei Erhalt neuer Objektdaten, also alle 2 s, durchgeführt werden. Neben dem eigentlichen Verfahren, welches eine kleinste umschließende Ellipse um alle Polygonpunkte der Sicherheitszonen bestimmt, liegt der Aufwand für die Berechnungen zudem in der Ermittlung der Überlappungssituationen zwischen den einzelnen Sicherheitszonen. Für den Test 5 sind das, bei  $n(n-1)/2$  durchzuführenden Tests, 1225 Überlappungstests. Eine mögliche Senkung der Rechenzeit könnte durch eine Verringerung der Polygonpunkte der Sicherheitszonen erfolgen. Durch die alleinige Einbeziehung der in Fahrzeugnähe befindlichen Hindernisse bzw. deren Sicherheitszonen zur Ermittlung der Hüllsicherheitszonen ist eine weitere Senkung der Rechenzeit möglich. So werden nur die Sicherheitszonen berücksichtigt, die sich innerhalb eines um das Fahrzeug liegenden Umkreises befinden. Hiermit kann die Rechenzeit zur Bestimmung der nächstliegenden Sicherheitszonen ebenfalls gesenkt werden.

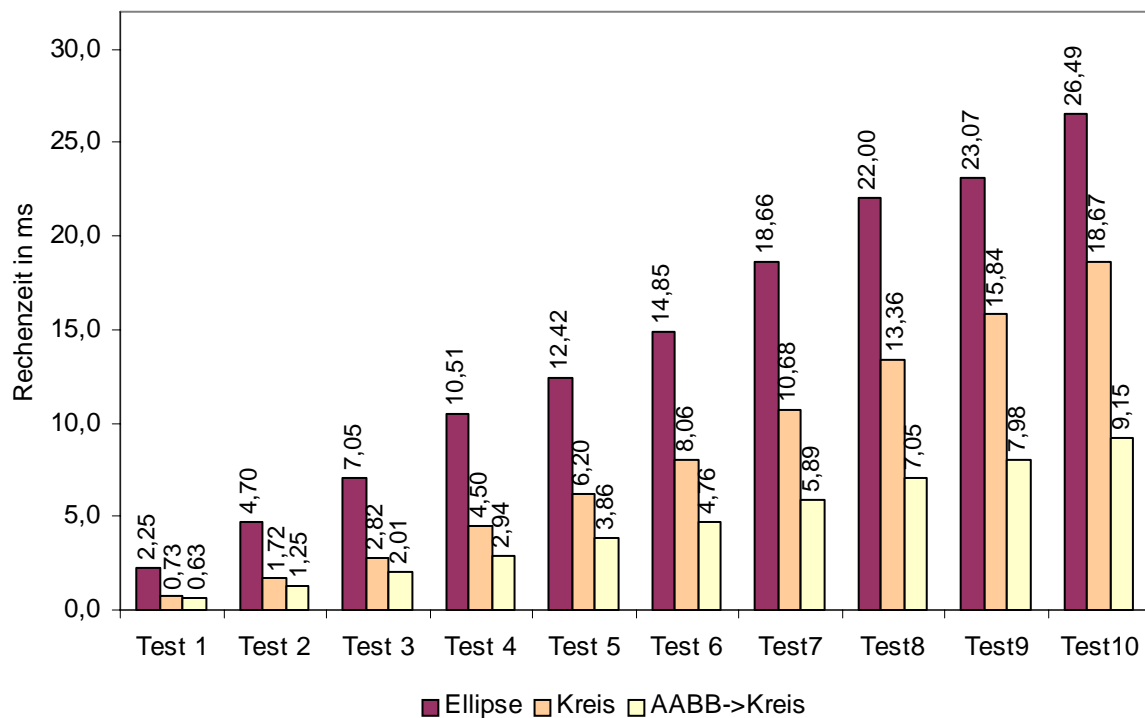


Abbildung 5.25 Ergebnisse der Laufzeittests zur Bestimmung der Hüllsicherheitszonen

Die Rechenzeiten der wichtigsten Komponenten des Verfahrens der Geometrischen Konstruktion sind in Abbildung 5.26 dargestellt. Die Rechenzeit zur Bestimmung des Gradienten ist konstant. Das liegt daran, dass maximal nur die drei nächstliegenden Hindernisse berücksichtigt werden. Der eigentliche Rechenaufwand besteht in der Berechnung der Abstände zu den Hindernissen und dem Sortieren der Abstände. Für Test 5 entspricht die Gesamt-rechenzeit des Verfahrens der Geometrischen Konstruktion 0,5 ms.

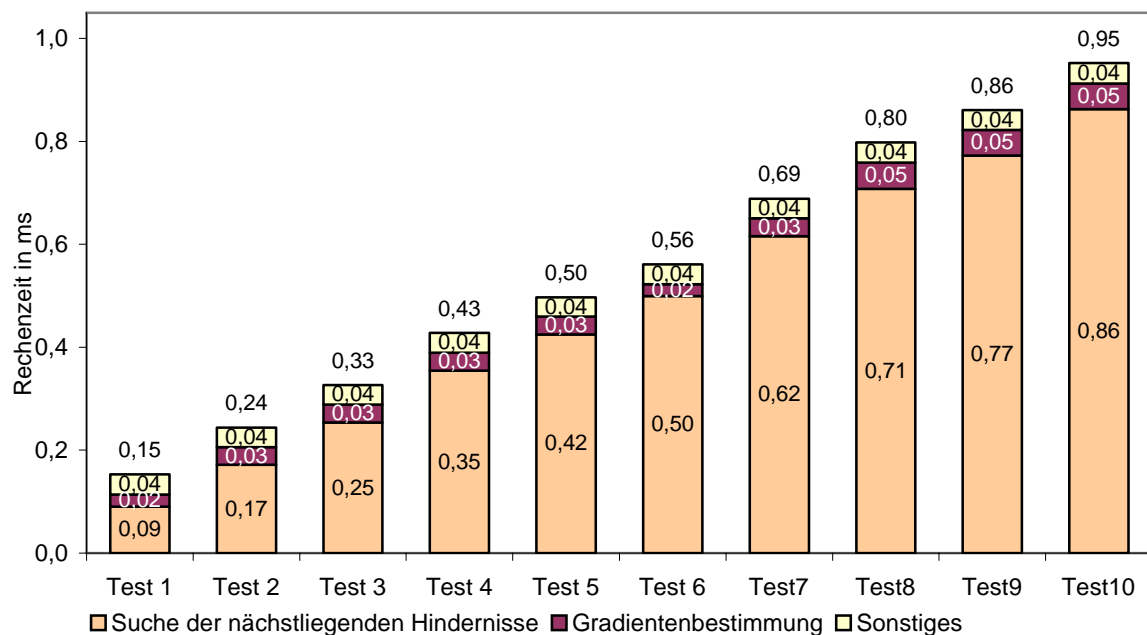


Abbildung 5.26: Rechenzeiten der einzelnen Komponenten des Verfahrens der Geometrischen Konstruktion

## 5.5 Zusammenfassung

Die Aufgabe der *Reaktiven Steuerung* im Modul *Fahrzeugführung in Sondersituationen* ist die Führung des Fahrzeuges zu einem vorgegebenen Zielpunkt unter Vermeidung aller Hinderniskollisionen. Dabei kann die *Reaktive Steuerung* diese Aufgabe als Teil des *Kollisionsvermeidungssystems* in Zusammenarbeit mit der *Wegeplanung* oder aber auch eigenständig lösen. So sind neben den Basisanforderungen an ein solches Verfahren, wie Echtzeitfähigkeit, Robustheit gegenüber den Objektdaten sowie die Berücksichtigung der Manövrierfähigkeit des Fahrzeuges, auch die Forderung nach einer möglichst optimalen Fahrtrajektorie gegeben.

Im einführenden Teil dieses Kapitels wurden einige aus der Literatur bekannte Verfahren vorgestellt und hinsichtlich ihres Einsatzes in einem Autonomen Unterwasserfahrzeug bewertet. Aufbauend auf dem Verfahren der Harmonischen Dipol-Potentiale (siehe Abschnitt 5.2.4) wurde ein neues Verfahren zur geometrischen Konstruktion von Gradientenlinien in Abschnitt 5.3 vorgestellt. Dieses Verfahren verbindet die Vorteile des Verfahrens der Harmonischen Dipol-Potentiale mit den Forderungen einer weg- und zeitoptimalen Fahrweise. Die Seeströmung sowie eine mögliche Geschwindigkeit der Hindernisse wurden bei der Konstruktion der Gradientenlinien durch Bildung einer Ersatzform für das Hindernis auf einfache Weise berücksichtigt.

Ein simulativer Vergleich der zu fahrenden Solltrajektorien des Verfahrens der Harmonischen Dipol-Potentiale mit dem Verfahren der Geometrischen Konstruktion in Abschnitt 5.4 zeigte die wesentlich geringeren Sollkursraten bei Verwendung des neu entwickelten Verfahrens. Dies ist für die Nutzung des Verfahrens bei den verwendeten Versuchsträgern AUV DeepC und AUV M63, die zur Schwergewichtsklasse der unbemannten Unterwasserfahrzeuge gehören [39], ein wichtiges Einsatzkriterium. Da der gebildete Gradient die Bewegungsreferenz des Fahrzeuges darstellt, ist der Einsatz einer Kursregelung zur Fahrzeugführung einer Gierwinkelregelung vorzuziehen. Das Verfahren der Geometrischen Konstruktion erfüllte die geforderten Echtzeitbedingungen bei den durchgeführten Performanceuntersuchungen.

## 6 Identifikationssystem

Bei der Durchführung einer Mission kann die Identifikation von Objekten eine geforderte Aufgabe sein. Eine Identifikation umfasst dabei eine visuelle oder kartographische Erfassung des Objektes und/oder eine Bestimmung seiner physikalischen/chemischen Eigenschaften. Das Modul *Missionsüberwachung* aktiviert dazu die Sensoren bzw. die Kamerasysteme und übergibt die Führung des Fahrzeuges an das Submodul *Identifikationssystem*. Hier werden aus den Vorgaben der Objektgeometrie und den Anforderungen des Fahrzeuges während der Identifikation (Objektabstand, Ausrichtung zum Objekt, ggf. Fahrgeschwindigkeit) spezielle Manöver zur Fahrzeugführung generiert. Die Führungsvorgaben der einzelnen Manöver werden an das Submodul *Führungsalgorithmen* zur Generierung von Sollvorgaben für den *Autopiloten* übergeben.

Im einführenden Teil dieses Kapitels findet eine Vorstellung von bekannten Anwendungen der autonomen Identifikation im Bereich der mobilen Systeme statt. Die Beschreibung der Anforderungen und die Aufgaben bei einer Objektidentifikation sowie eine Erläuterung des neu entwickelten Lösungskonzeptes erfolgen im Anschluss. Dabei werden die Handlungsabläufe der einzelnen Identifikationsaufgaben im Detail vorgestellt und anhand von Simulationen erläutert.

### 6.1 Anwendungen der Objektidentifikation

Dieser Abschnitt soll einige von autonomen mobilen Systemen zu lösende Identifikationsaufgaben vorstellen. Es wird kein Anspruch auf Vollständigkeit erhoben, sondern ein Abriss von technisch realisierten und zukünftigen Anwendungen aufgezeigt.

Das autonome mobile System MOSRO 1 ist der erste serienreife mobile Sicherheitsroboter der Welt [83]. Seine Aufgabengebiete liegen in der Objektüberwachung. Er stellt eine Alternative zum Menschen bei einer gefährlichen oder zu kostenintensiven Überwachung dar. Über spezielle Sensoren kann er Gefahren wie Gas, Rauch und Temperaturerhöhung detektieren und an eine Zentrale weitermelden. Menschliche Konturen und Bewegungen kann er auch in absoluter Dunkelheit und durch Wände erkennen. Entdeckte Personen werden aufgefordert, sich über ihren Fingerabdruck zu identifizieren. Erfolgt dies nicht, löst er einen Alarm aus. Ein interaktives mobiles Servicesystem für einen Baumarkt wird in [19] vorgestellt. Dieses System soll den Kunden beim Einkauf beratend zur Seite stehen. In Analogie zu seinen menschlichen Vorbildern soll der Roboter selbstständig Kunden kontaktieren, um seine Hilfe anzubieten. Zu diesem Zweck ist eine Aufmerksamkeitsteuerung implementiert, die auf der Basis visueller und akustischer Informationen eine Personenerkennung durchführt. Zur Kontaktaufnahme wird ein vorgegebener Ablauf abgearbeitet (Personenverifikation → Anfahrt zum Kunden → Begrüßung des Kunden).

Für einen autonomen Helikopter wird in [123] ein System für Fassadenaufnahmen von Gebäuden entwickelt. Dieses System kann zur Generierung eines 3D-Modells für eine virtuelle Realität oder zu visuellen Sicherheitsinspektionen eingesetzt werden. Um die einzelnen Gebäude in den richtigen Positionen (Fassadenseiten) zu fotografieren, befindet sich ein GIS-Programm (Geographisches Informationssystem) an Bord. Die Wegeplanung überprüft jede einzelne Aufnahmeposition auf Erreichbarkeit und bestimmt eine optimale Folge zum Abfliegen der einzelnen Positionen. So besteht die Missionsaufgabe aus einer Sequenz von Anflugmanövern zu den einzelnen Positionen und der Ausrichtung des Helikopters zur Aufnahme der Gebäudefassaden.

Für das AUV „Tri-Dog 1“ erfolgt die Strukturuntersuchung von Wänden unter Verwendung eines Messsystems bestehend aus einer CCD Kamera und einem Laser [95]. Dieses System wird zur Ermittlung von Objektformen und zur Fahrzeugführung eingesetzt. Einsatzfälle für solche Strukturuntersuchungen liegen in der Zustandsbestimmung von Unterwasserbauten (Träger, Caisson (Senkkasten), Kaimauern) in Hafenanlagen, von Dämmen oder Schiffsrümpfen. Ein praktischer Einsatzfall des AUVs „Tri-Dog 1“ war die Inspektion einer Wellenbrecherwand und ihrer Felsaufschüttungen in der Kamaishi Bucht in Japan [94]. Bei der Mission wurden mäanderförmig Wegpunkte entlang der Wand auf dem Seeboden definiert, die das Fahrzeug unter Verwendung eines Bewegungsreglers abfuhr wobei es Fotoaufnahmen erstellte. In [116] wird ein System zur Kabelverfolgung vorgestellt. Die Fahrzeugsteuerung erfolgt dabei in einer Kombination aus mehreren Führungsalgorithmen. Die Erfüllung der Aufgabe ist in vier Phasen klassifiziert. Die erste Phase umfasst das Aussetzen und das Führen des Fahrzeuges in Richtung der Ziellinie mit einer Ziellinienregelung (engl.: Line-of-Sight Guidance (LOS) [58]). In der zweiten Phase folgt das Fahrzeug dem Zielrichtungswinkel mit einer Wegpunktregelung mit maximaler Geschwindigkeit. Hierbei wird das Fahrzeug in die Nähe des Zielgebietes geführt. In der vorletzten Phase muss das Fahrzeug in Zielnähe seine Geschwindigkeit verringern und abtauchen, um sich zum Kabel auszurichten, bevor die letzte Phase der Kabelverfolgung durchgeführt werden kann.

Die US Navy sieht in ihrem Unmanned Undersea Vehicle (UUV) Master Plan ein großes Potential für die Objektidentifikation bei der Minenbekämpfung [39]. Bei dieser Anwendung können AUVs die Minen in einer mehrstufigen Strategie detektieren, klassifizieren und identifizieren. Andere Einsatzfelder von Identifikationsaufgaben liegen im Bereich des Anti-Terrorkampfes und des Zivilschutzes (engl.: homeland security). Aufgabengebiete sind Objektuntersuchung und -lokalisierung an Schiffsrümpfen, an Anlegestegen und Kaianlagen.

Zur Lokalisierung von hydrothermischen Quellen oder Süßwasserquellen im Meer können koordinierende Gruppen von AUVs eingesetzt werden. Süßwasserquellen erzeugen eine räumlich ausgedehnte Wolke mit einem variablen Salzgehalt, der mit zunehmender Annäherung an die Quelle abnimmt. In [105] wird eine solche Aufgabe in einer Konfiguration aus einem autonomen Überwasserfahrzeug und drei AUVs gelöst. Die Verwendung des Überwasserfahrzeuges erlaubt eine absolute Positionierung, wodurch ein exaktes Folgen der vorgegebenen Route möglich ist. Die Gruppe der Unterwasserfahrzeuge detektiert die Süßwasserwolke und folgt ihr unter Verwendung der bei den einzelnen Fahrzeugen gemessenen Wasserproben und der Formationsgeometrie der Fahrzeuge zueinander. Dabei erfolgt die Fahrzeugsteuerung eines AUVs für die Vorwärtsgeschwindigkeit und die Winkelgeschwindigkeit in Abhängigkeit des Soll- und des Istabstandes und der Soll- und Istrichtung zu den anderen AUVs sowie der gemessenen Salzwasserkonzentration.

## 6.2 Konzept der Objektidentifikation

### 6.2.1 Anforderungen und Aufgaben

Bei den durchgeführten Missionen des AUVs DeepC umfasst die Identifikationsaufgabe eine visuelle Aufzeichnung der vom Sonar detektierten Objekte. Dabei besteht die Aufgabe des *Identifikationssystems* darin, das Fahrzeug zum Objekt zu führen und es während der Kameraaufnahmen entsprechend auszurichten. Dies soll auch bei einer vorhandenen Seeströmung sicher funktionieren. Die Manövrierfähigkeit des Fahrzeuges ist jedoch in dieser Situation eingeschränkt.

Die Verfahren von Visual Servoing (bildbasierte Regelung) können zur Erfüllung dieser Positionieraufgabe nicht genutzt werden, was nachfolgend kurz erläutert werden soll. Beim Visual Servoing verwendet der Regler bildbasierte Informationen zur Bewegung des Fahrzeuges. Ein Vorteil dieses Ansatzes besteht im robusteren Verhalten gegenüber Modellierungsfehlern, im Vergleich zu einer positionsgeführten Regelung [137]. Dieses Verfahren setzt als Sollwert ein Bild voraus, das in der Ziel-Lage aufgenommen wurde. Solche Informationen können jedoch nicht bereitgestellt werden, wenn die zu identifizierenden Objekte unbekannt sind. Zusätzlich können durch schlechte Sichtverhältnisse die Ist-Bildmerkmale für die Regelung nicht zur Verfügung stehen, wodurch eine Fahrzeugführung nicht mehr möglich ist.

Für die Manövrierung des Fahrzeuges zum Objekt werden deshalb ausschließlich die Informationen der von der *Objekterkennung* generierten, elliptischen Zylinderformen verwendet. Die Sonarrohinformationen (minimaler Abstand zum Objekt, Objektpeilung) können aufgrund ihrer geringen Updateraten nicht genutzt werden, da das *Identifikationssystem* zyklisch in jeder Updateroutine des Moduls *FIS* abgearbeitet wird, um die Werte zur Fahrzeugführung zu generieren.

### 6.2.2 Vorstellung des entwickelten Konzeptes

Die Verwendung der aufbereiteten Objektinformationen vom Modul *Objekterkennung* macht eine einfache und robuste Führung des Fahrzeuges während einer Identifikation möglich. Die Manövrierung des Fahrzeuges erfolgt so unter Nutzung der geometrischen Daten eines elliptischen Zylinders. In Abhängigkeit von den vorgegebenen Objektabmaßen werden die Arbeitsmodi *Ausrichten auf ein Objekt* und *Umfahren eines Objektes* unterschieden. Die einzelnen Modi setzen sich aus mehreren Manövern bzw. Führungsaufgaben zusammen, die abhängig von der Seeströmung und der Fahrzeugposition zum zu identifizierenden Objekt aktiv sind. Abbildung 6.1 zeigt das Strukturbild des Identifikationssystems. Die Aufgabe der Manöver ist die Generierung von Vorgaben für die einzelnen Führungsalgorithmen aus den Navigations- und Objektdaten. Nach Abschluss einer Identifikation wird ein anzufahrender Zielpunkt auf der aktuellen Missionsroute durch das Submodul *Zielpunktgenerierung* ermittelt. Dieser Zielpunkt wird dann durch das intern aktivierte *Kollisionsvermeidungssystem* angefahren. So kann das Fahrzeug sicher um das Objekt geführt werden, um zur Missionsroute zurückzukehren. Werden bei der Rückfahrt zur Missionsroute weitere Objekte detektiert, so erfolgt auch deren Identifikation, wenn sie noch nicht als identifiziert gekennzeichnet sind. Hierzu existiert eine Liste mit den bereits identifizierten Objekten. In den nächsten Abschnitten werden beide Modi *Ausrichten auf ein Objekt* und *Umfahren eines Objektes* vorgestellt.

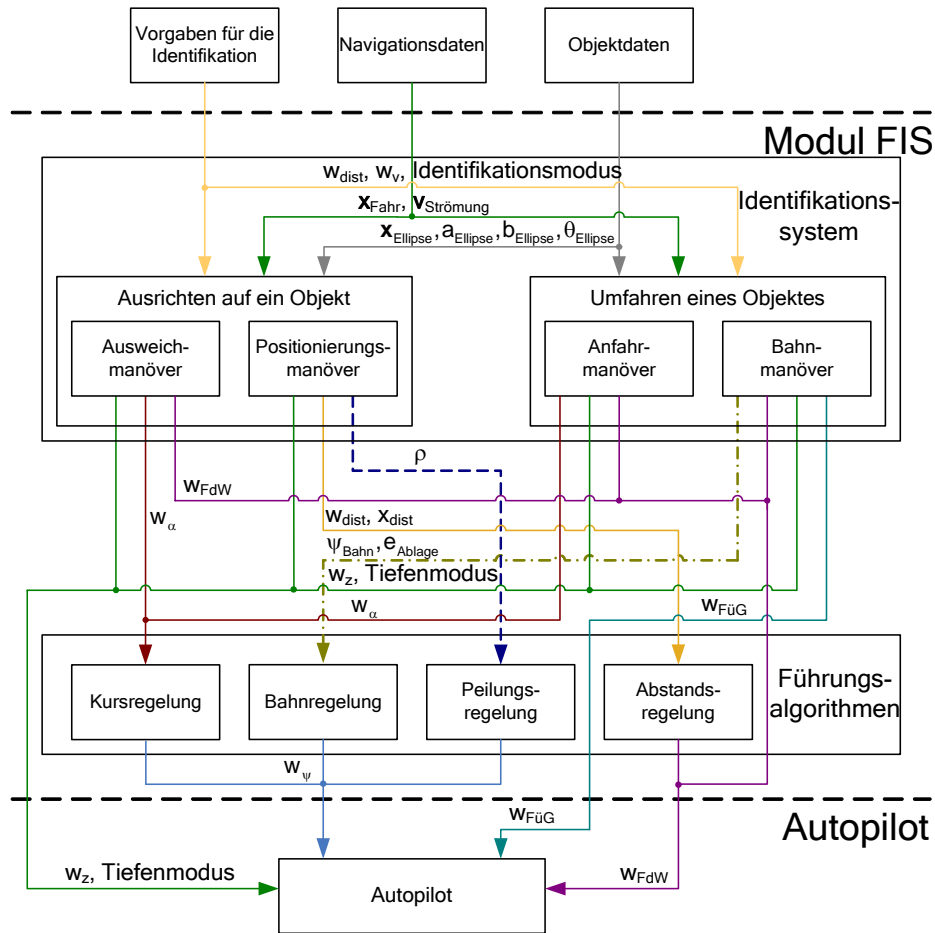


Abbildung 6.1: Struktur des Identifikationssystems

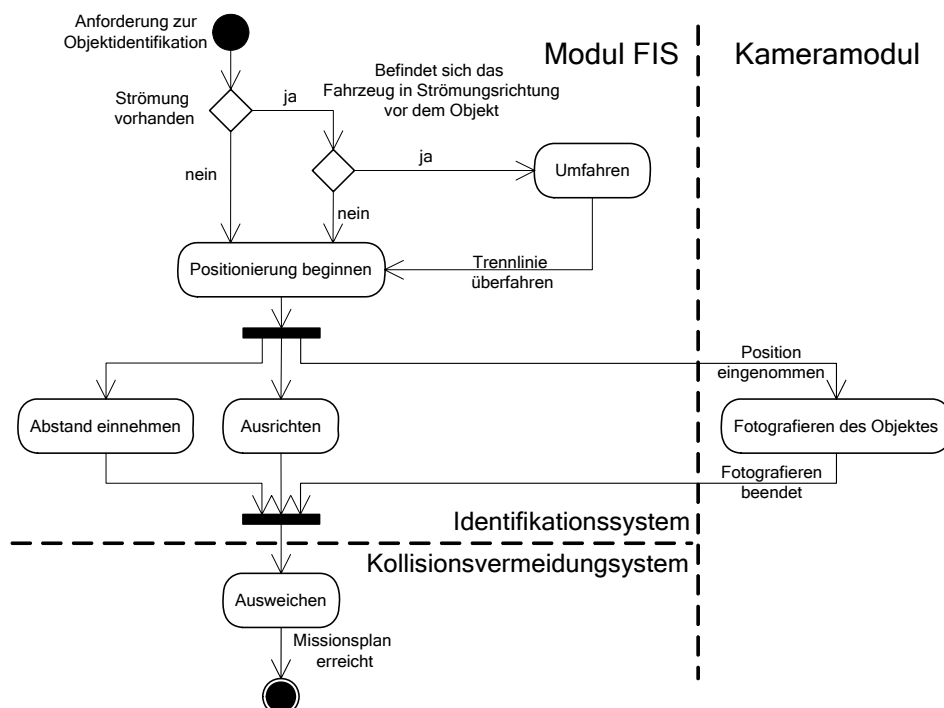
## 6.3 Ausrichten auf ein Objekt

In diesem Abschnitt werden die Algorithmen zur Führung des Fahrzeuges im Identifikationsmodus *Ausrichten auf ein Objekt* vorgestellt. Dabei erfolgt eine Beschreibung des Ablaufes der Identifikation mit den dazugehörigen Manövern. Die dafür notwendigen geometrischen Berechnungen werden im Detail erläutert.

### 6.3.1 Ablauf der Identifikation

Für die visuelle Aufnahme des zu identifizierenden Objektes soll das Fahrzeug eine definierte Lage zum Objekt einnehmen und während der Identifikation beibehalten. Zur Erfüllung dieser Aufgabe stehen als Steuergrößen für den externen *Autopiloten* nur der Gierwinkel und die Fahrgeschwindigkeit zur Verfügung (siehe Kapitel 8), wodurch sich das Positioniermanöver aus zwei Teilaufgaben zusammensetzt. Dazu wird eine Position  $x_{fix}$  auf dem elliptischen Zylinder vorgegeben, auf den sich das Fahrzeug ausrichtet. Im ausgerichteten Zustand kann das Fahrzeug einen definierten Abstand zum Objekt durch Variation seiner Fahrgeschwindigkeit einnehmen. Das Positioniermanöver berechnet also einen Peilungswinkel  $\rho$  für die Peilungsregelung und einen Istabstand  $x_{dist}$  für die Abstandsregelung der *Führungsalgorithmen*. Die definierte Position  $x_{fix}$  ist die dem Fahrzeug Nächstgelegene. Die Berechnung dieser Position wird in Abschnitt 6.3.2 beschrieben. Abbildung 6.2 stellt das Aktivierungsdiagramm bei einer Identifikation dar.



Abbildung 6.2: Aktivierungsdiagramm im Identifikationsmodus *Ausrichten auf ein Objekt*

Tritt in diesem Modus eine starke Seeströmung auf, kann sich das Fahrzeug nur in entgegen der Strömung gerichteter Lage sicher positionieren. Dazu muss das Fahrzeug in den Strömungsschatten des Objektes geführt werden, um eine Position einnehmen und halten zu können. Das Umfahren des Objektes erfolgt mit den im Abschnitt 5.3 vorgestellten Verfahrens der Geometrischen Konstruktion. Zur Aktivierung der einzelnen Manöver werden, unter Beachtung der Strömungsrichtung und eines Sicherheitsabstandes, Gebiete um das Objekt definiert, wie sie in Abbildung 6.3 dargestellt sind. Im unteren Teil der Abbildung befindet sich das Fahrzeug im Ausweichmodus und umfährt das Objekt unter Vorgabe eines im Strömungsschatten gelegenen Zielpunktes. Wenn das Fahrzeug eine vorgegebene Trennlinie überschritten hat, erfolgt die Positionierung zum Objekt.

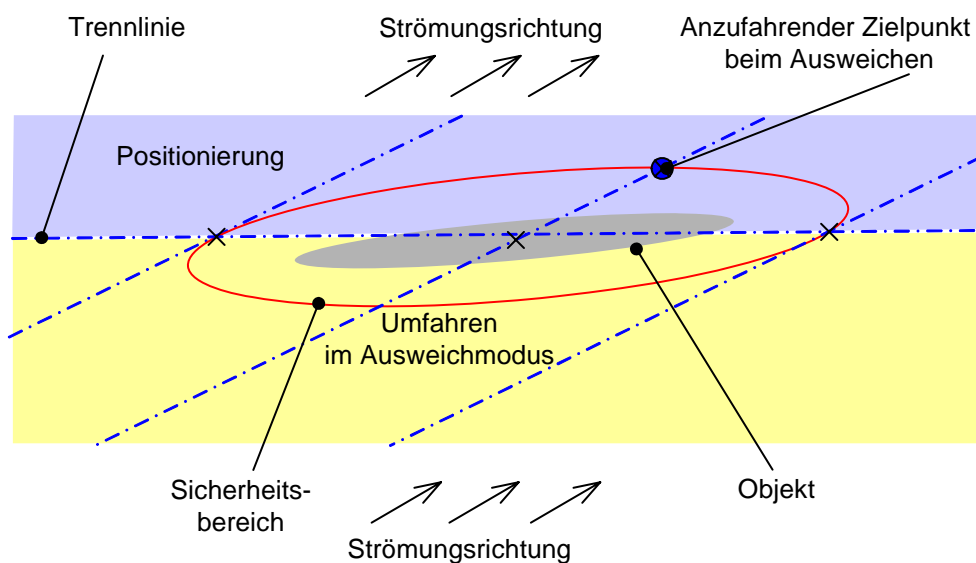


Abbildung 6.3: Ausrichten auf ein Objekt in Abhängigkeit der Strömung

### 6.3.2 Bestimmung des nächstgelegenen Punktes auf der Ellipse

In diesem Abschnitt erfolgt die Herleitung des zum Fahrzeug nächstgelegenen Punktes auf dem elliptischen Zylinder. Durch die achsenparallele z-Ausrichtung des elliptischen Zylinders werden die Berechnungen in der Horizontalebene zwischen einer Ellipse und einem Punkt durchgeführt. Zur Vereinfachung der Berechnungen wird die Ellipse achsenparallel in den Koordinatenursprung gelegt. Die Ellipse kann so durch ihre Mittelpunktsleichung beschrieben werden:

$$Q(x, y) = \left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 - 1. \quad (6.1)$$

Die Position des Punktes  $\mathbf{x}_{Fahr} = [u \ v]$  (Fahrzeugposition) wird dann in Analogie zum Abschnitt 5.3.3 durch eine Translation um den Mittelpunkt der Ellipse  $\mathbf{x}_{Ellipse}$  verschoben und durch eine Rotation um den Rotationswinkel der Ellipse  $\theta_{Ellipse}$  achsenparallel ausgerichtet. Der nächstgelegene Punkt auf der Ellipse  $\mathbf{x}_{nah} = [x \ y]$  muss die Bedingung erfüllen, dass die Strecke  $\mathbf{x}_{Fahr} - \mathbf{x}_{nah}$  eine Normale der Ellipse ist [42]. Die Richtung dieser Normalen  $\mathbf{n}$  lässt sich durch Bildung des Gradienten  $\nabla Q(x, y)$  der Gleichung (6.1) bestimmen. Abbildung 6.4 soll die oben beschriebenen geometrischen Zusammenhänge noch einmal verdeutlichen. Die Gleichung der Strecke lautet damit:

$$\mathbf{x}_{Fahr} - \mathbf{x}_{nah} = t \nabla Q(x, y) = t \begin{bmatrix} 2 \frac{x}{a^2} \\ 2 \frac{y}{b^2} \end{bmatrix}. \quad (6.2)$$

Durch Umstellen der Formel (6.2) nach  $\mathbf{x}_{nah}$ :

$$\mathbf{x}_{nah} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{ua^2}{(t+a^2)} \\ \frac{vb^2}{(t+b^2)} \end{bmatrix} \quad (6.3)$$

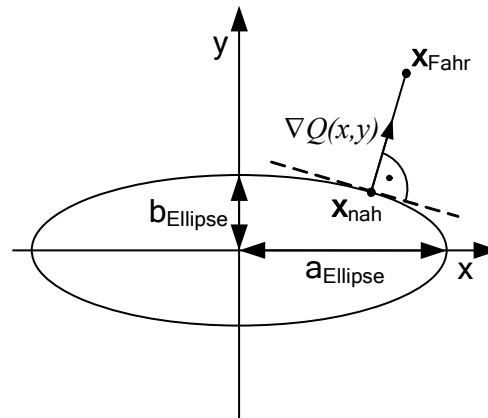
und Einsetzen in Gleichung (6.1) kann die Variable  $t$  durch Nullstellenbestimmung eines Polynoms 4. Ordnung:

$$p_4 t^4 + p_3 t^3 + p_2 t^2 + p_1 t + p_0 = 0 \quad (6.4)$$

mit den Parametern:

$$\begin{aligned} p_0 &= a^4 b^4 - a^4 b^2 v^2 - b^4 a^2 u^2 \\ p_1 &= 2.0 \left( a^2 b^4 + a^4 b^2 - a^2 b^2 (u^2 + v^2) \right) \\ p_2 &= a^4 + b^4 + 4a^4 b^4 - a^2 u^2 - b^2 v^2 \\ p_3 &= 2.0 (a^2 + b^2) \\ p_4 &= 1.0 \end{aligned} \quad (6.5)$$

ermittelt werden. Jede gefundene Nullstelle von Gleichung (6.4) wird in Gleichung (6.3) eingesetzt und der Abstand  $\|\mathbf{x}_{Fahr} - \mathbf{x}_{nah}\|$  bestimmt. Die Position, mit der der kürzeste Abstand ermittelt wird, entspricht dann dem gesuchten Wert  $\mathbf{x}_{nah}$ .

Abbildung 6.4: Nächster Punkt  $x_{nah}$  auf einer Ellipse zu einem gegebenen Punkt  $x_{Fahr}$ 

Die so bestimmte Position  $x_{nah}$  wird anschließend wieder vom Bildbereich in den Originalbereich rücktransformiert (siehe auch Abschnitt 5.3.3).

### 6.3.3 Test des Verfahrens

Die Überprüfung des Identifikationsmodus *Ausrichten auf ein Objekt* bei einer vorhandenen Seeströmung wird exemplarisch anhand einer Simulink-Simulation für das AUV DeepC durchgeführt. Die Startposition des Fahrzeuges wird so gewählt, dass sie in Strömungsrichtung vor dem zu identifizierenden Objekt liegt. Im ersten Teil des Identifikationsmanövers umfährt das Fahrzeug das Objekt unter Verwendung des Verfahrens der Geometrischen Konstruktion. Durch die Verwendung einer Kursregelung zum Folgen der Gradientenlinien, wie sie in Abschnitt 5.4.2 favorisiert wurde, kann das Fahrzeug entlang der Tangente des Sicherheitsbereiches geführt werden. Überfährt das Fahrzeug die definierte Trennlinie (siehe Abbildung 6.3), erfolgt die Positionierung zum Objekt. Durch die Ausrichtung des Fahrzeuges zum nächstgelegenen Punkt auf dem Objekt in Kombination mit der am Fahrzeug angreifenden Strömung richtet sich das Fahrzeug entgegen der Strömungsrichtung zum Objekt aus und beschreibt eine so genannte „Hundekurve“ [13]. Der Sollabstand zum Objekt wird durch die Variation der Fahrgeschwindigkeit der aktivierten Abstandsregelung erreicht.

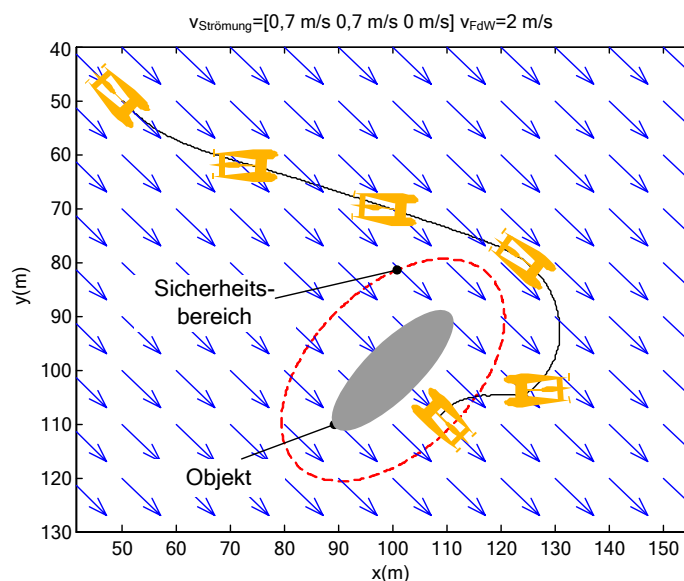


Abbildung 6.5: Trajektorienverlauf des Identifikationsmanövers bei Seeströmung

## 6.4 Umfahren eines Objektes

Bei großräumigeren Objekten können nicht alle Objektinformationen durch die Kameraaufnahmen von einer Position aus erfasst werden. In einer solchen Situation muss das Fahrzeug das zu identifizierende Objekt in einem definierten Abstand und in einer vorgegebenen Lage umfahren, um kontinuierlich visuelle Aufnahmen des Objektes aufzuzeichnen. Dieser Identifikationsmodus wird als *Umfahren eines Objektes* bezeichnet und soll in diesem Abschnitt vorgestellt werden.

### 6.4.1 Ablauf der Identifikation

Das Umfahren eines Objektes erfolgt in drei Phasen. Die erste Phase ist aktiv, wenn der Abstand des Fahrzeuges zur nächstgelegenen Position auf dem Objekt  $x_{fix}$  größer als ein vorgegebener maximaler Abstand  $x_{dist\_max}$  ist. In dieser Phase fährt das Fahrzeug mit aktivierter Kursregelung auf die Position  $x_{fix}$  zu. Die Berechnung der Position  $x_{fix}$  erfolgt unter Verwendung der geometrischen Beziehungen des Abschnittes 6.3.2. Nach Unterschreitung des vorgegebenen maximalen Abstandes  $x_{dist\_max}$  beginnt in der zweiten Phase das Einfahren auf die vorgegebene Sollbahn mit der Bahnregelung. Unterschreitet die Sollbahnablage über einen definierten Zeitraum einen vorgegebenen Wert, kann davon ausgegangen werden, dass der Einschwingvorgang abgeschlossen ist. Hier wird die dritte Phase aktiviert, in der das Fahrzeug in einem vorgegebenen Abstand das Objekt umfährt. Dieser Zustand wird an das Modul *Missionsüberwachung* gemeldet, das seinerseits das Kameramodul aktiviert. Abbildung 6.6 zeigt die Fahrtrajektorie beim Umfahren eines Objektes mit den einzelnen Phasen. Die Umfahrrichtung ist abhängig von der Seeströmung. Um eine bessere Stabilität beim Einfahren zu erreichen, soll das Fahrzeug entgegen der Strömungsrichtung auf die Sollbahn geführt werden. Die Bestimmung der Ablage erfolgt in Abhängigkeit von der Umfahrrichtung durch die folgende Gleichung:

$$e_{Ablage} = \begin{cases} w_{dist} - x_{dist} & \text{für Umfahren im Uhrzeigersinn} \\ x_{dist} - w_{dist} & \text{für Umfahren entgegen dem Uhrzeigersinn} \end{cases} \quad (6.6)$$

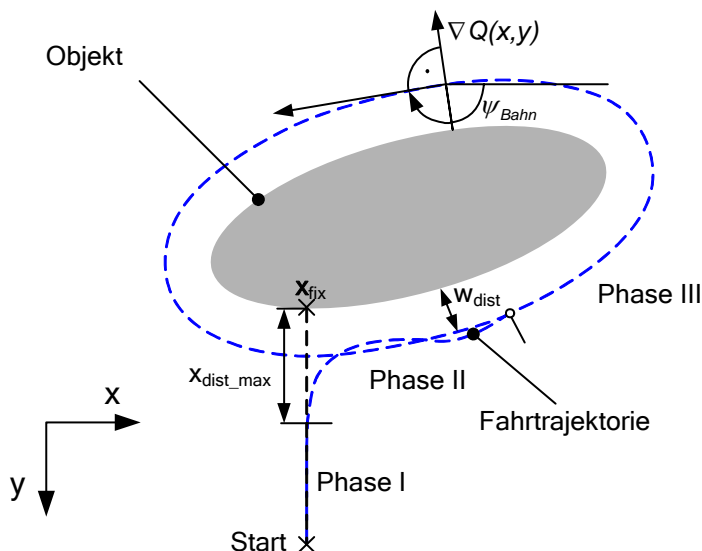
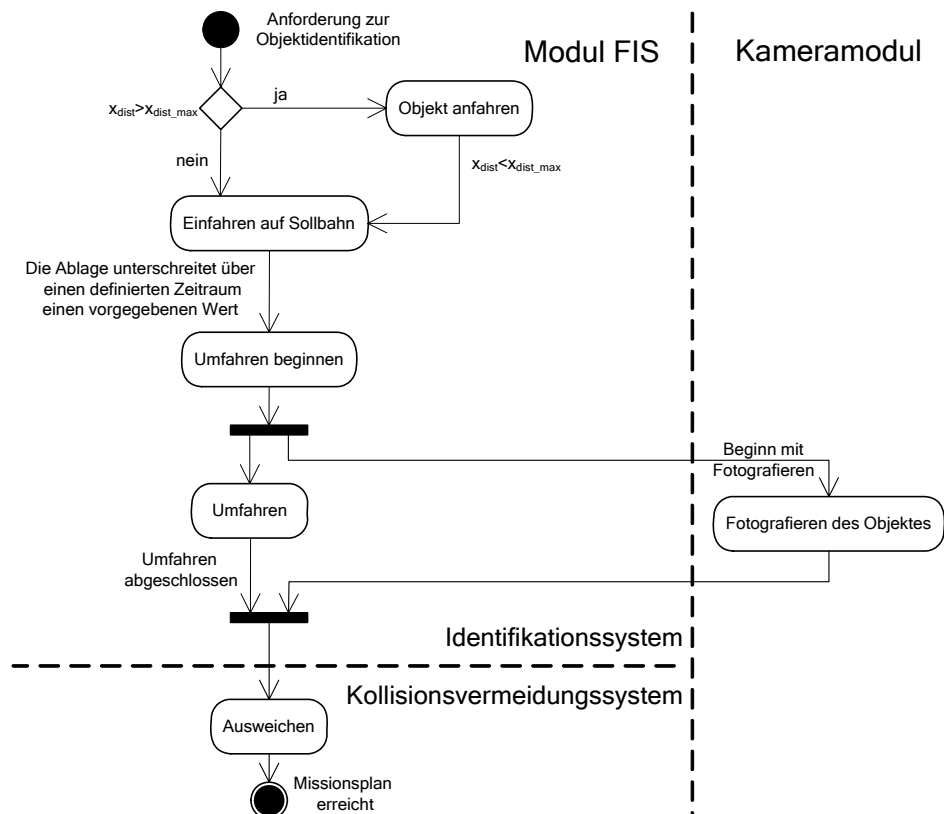


Abbildung 6.6: Fahrtrajektorie der drei Phasen beim Umfahren eines Objektes

Abbildung 6.7: Aktivierungsdiagramm im Identifikationsmodus *Umfahren eines Objektes*

Die Richtung des Vektors der aktuellen Bahnrichtung  $\psi_{Bahn}$  liegt orthogonal (senkrecht) zum Gradienten  $\nabla Q(x, y)$  der elliptischen Grundfläche des Objektes, der durch die aktuelle Position  $x_{fix}$  bestimmt wird (siehe auch Abbildung 6.6).

Das Aktivierungsdiagramm für das *Umfahren eines Objektes* ist in Abbildung 6.7 dargestellt. Die Abbildung 6.8 zeigt die Aktivität der einzelnen Manöver im Identifikationsmodus *Umfahren eines Objektes* in Abhängigkeit der Fahrzeugposition vom Objekt.

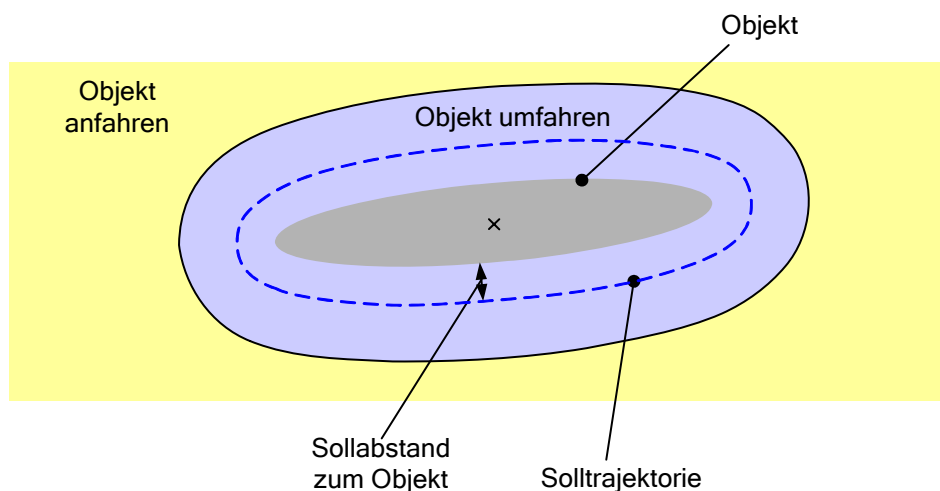


Abbildung 6.8: Umfahren eines Objektes

## 6.4.2 Test des Verfahrens

Die Überprüfung des Identifikationsmodus *Umfahren eines Objektes* erfolgt durch eine Simulink-Simulation für das AUV M63. Dabei wird eine Seeströmung mit einem Betrag von 1 m/s und einer Richtung von  $-45^\circ$  vorgegeben. Während des ersten Manövers fährt das Fahrzeug auf direktem Weg auf das zu identifizierende Objekt zu (siehe Abbildung 6.9). Bei diesem Manöver ist die Kursregelung aktiv. Es ist deutlich die Ausrichtung des Fahrzeuges entgegen der Strömungsrichtung zu erkennen, die notwendig ist, um den Sollkurs bei der vorhandenen Strömung zu halten. Das Fahrzeug fährt entgegen der Strömungsrichtung auf die Solltrajektorie ein, wodurch der Einschwingvorgang schnell beendet ist. Beim Umfahren des Objektes entsteht eine maximale Ablage von 2 m. Dies ergibt sich aus der variablen Bahnrichtung, verbunden mit der veränderlichen Anströmung des Fahrzeuges während des Umfahrens und der Auslegung des Bahnreglers für Störung und Führung. Das verwendete Konzept der Bahnregelung kompensiert über den integralen Anteil die angreifende Strömung. Das Einbeziehen der Strömungsinformation und des zukünftigen Verlaufes der Bahnrichtung in den Regelungsentwurf könnte zu einer Verbesserung führen. Eine weitere Ursache für die Ablage ist durch die begrenzte Drehgeschwindigkeit des Fahrzeuges gegeben, wodurch das Folgen von schnellen Bahnrichtungsänderungen bei starken Bahnkrümmungen nicht möglich ist. An dieser Stelle sei angemerkt, dass durch den minimalen Wenderadius der verwendeten Klasse von AUVs kleinere Objekte nicht umfahren werden können.

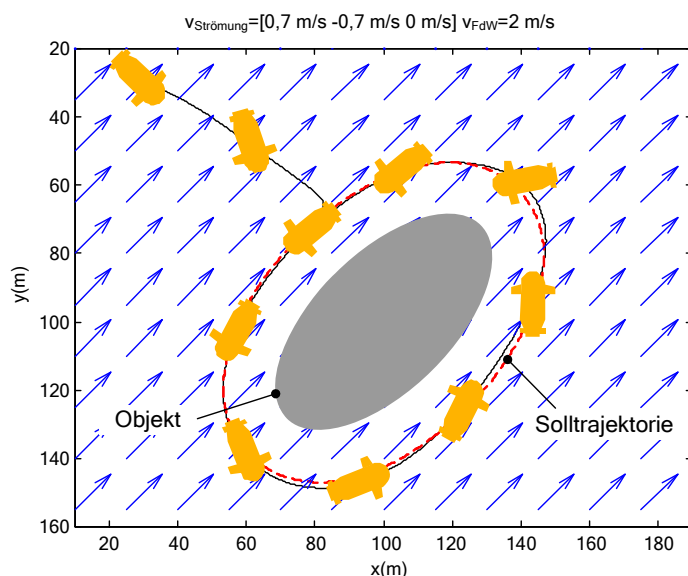


Abbildung 6.9: Trajektorienverlauf des Identifikationsmanövers *Umfahren eines Objektes* bei Seeströmung

## 6.5 Zusammenfassung

Dieses Kapitel beschreibt die neu entwickelten Algorithmen zur Führung des Fahrzeuges bei den Identifikationsaufgaben *Ausrichten auf ein Objekt* und *Umfahren eines Objektes*. Die Erfüllung der beiden Aufgaben erfolgt durch die Abarbeitung einzelner Manöver. Diese Manöver werden in Abhängigkeit der Fahrzeugposition zum Objekt aktiviert und generieren die Vorgaben für die Führungsalgorithmen. Bei den simulativen Untersuchungen konnten die Verfahren bei einer vorhandenen Seeströmung erfolgreich getestet werden. Die Verwendung eines elliptischen Hüllkörpers zur Abdeckung des Objektes ermöglicht eine einfache und robuste Manövrierung des Fahrzeuges während der Identifikation.

## 7 Zielpunktgenerierung

Dieses Kapitel stellt die entwickelten Algorithmen des Submoduls *Zielpunktgenerierung* vor. Die *Zielpunktgenerierung* berechnet während einer Sondersituation (Kollisionsvermeidung oder Objektidentifikation) einen Rendezvouspunkt mit dem abzufahrenden Missionsplan unter Verwendung seiner Basismanöver und der detektierten Objekte. Dieser Rendezvous- bzw. Zielpunkt wird so berechnet, dass möglichst wenig Verluste an durchzuführenden Missionsaufgaben infolge nicht befahrener Manöverelemente entstehen und das Fahrzeug in Abhängigkeit von seinen Manövrierfähigkeiten kollisionsfrei auf die Missionsroute einfahren kann. Ausgehend von einigen Anwendungen für eine Zielpunktvorgabe erfolgt eine detaillierte Beschreibung des neu entwickelten Verfahrens zur Zielpunktgenerierung.

### 7.1 Anwendungen zur Zielpunktgenerierung

Eine Zielpunktvorgabe ist für die autonome Bewegung eines mobilen Roboters bei der Kollisionsvermeidung bzw. den so genannten Punkt-zu-Punkt-Bewegungsaufgaben erforderlich [71]. Dabei durchfährt der Roboter ein unbekanntes oder teilweise bekanntes Operationsgebiet und versucht, unter Verwendung seines globalen Hinderniswissens und seiner Umweltsensoren, kollisionsfrei zum vorgegebenen Zielpunkt zu gelangen. In vielen wissenschaftlichen Arbeiten zu diesem Thema wird dieser Zielpunkt beim Start der Mission mit übergeben und so als bekannt vorausgesetzt. Bei semi-autonomen Missionen, wie beim NASA Mars Robotic Program wird ein anzufahrender Zielpunkt von einer Bodenstation an den Mars Rover übergeben [37]. Die Bestimmung des Zielpunktes erfolgt auf Basis einer visuellen Terrainausswertung und des gesammelten Kartenmaterials. Eine direkte Führung des Rovers ist durch die Laufzeitverzögerung der gesendeten Kommandos und der empfangenen Bilder sowie durch Phasen, bei denen es keine Kommunikationsmöglichkeit gibt, nicht möglich.

Die ersten mobilen Systeme zur Unterstützung eines Soldaten im Kampfeinsatz wurden ausschließlich teleoptisch geführt. Solche Systeme nahmen die volle Aufmerksamkeit des Operators in Anspruch und verhinderten so die Aufrechterhaltung seines eigenen Schutzes. Aktuelle Systeme umfassen eine Wegpunktnavigation und ein Kollisionsvermeidungssystem [28]. Der Operator kann über eine Bedieneinheit Zielpunkte vorgeben, die das mobile System eigenständig unter Vermeidung von Kollisionen anfährt.

In [71] wird zur Aktualisierung des globalen Wegeplanes durch neu erworbenes Hinderniswissen das Verfahren der Mehrfachwellenbearbeitung eingesetzt. Ausgehend von einem Zielpunkt ZP und einem Startpunkt SP erfolgt eine gleichzeitige wellenförmige Bearbeitung des Operationsgebietes. Beim Zusammentreffen der beiden Wellen wird die Bearbeitung abgebrochen und der Treffpunkt TP1 der beiden Wellen als Zwischenzielpunkt ZZP1 abgespeichert. In einem zweiten Planungsschritt werden von diesem Zwischenzielpunkt ZZP1 und dem Startpunkt SP aus zwei neue Bearbeitungswellen generiert (siehe Abbildung 7.1).

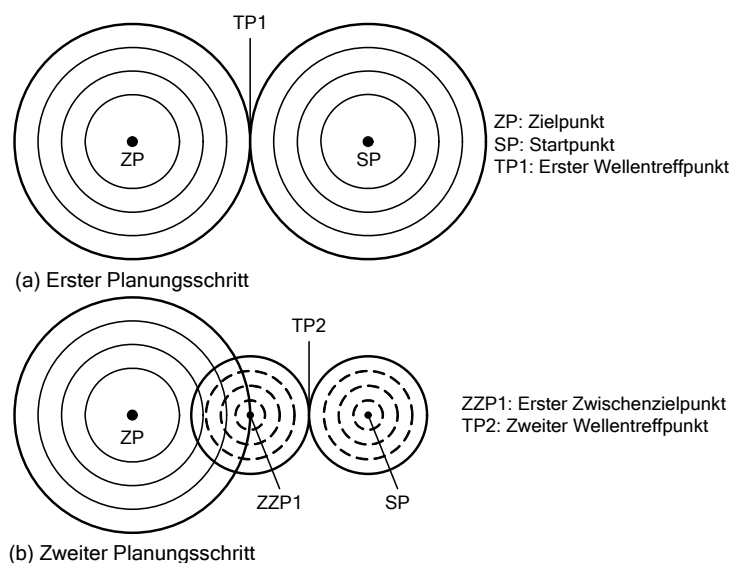


Abbildung 7.1: Mehrfachwellen zur systematischen Bearbeitung des Operationsgebietes [71]

Dieses Verfahren kann nach dem oben beschriebenen Vorgehen durch weitere Planungsschritte verfeinert werden. Vor jedem Planungsschritt wird überprüft, in welchen Bereichen des Operationsgebietes sich das Hinderniswissen seit dem letzten Planungsschritt verändert hat. Nur in diesen Bereichen findet eine Neuplanung zum Zwischenzielpunkt statt [71]. Durch dieses Vorgehen erfolgt eine ständige Aktualisierung des Planes, ohne dass das gesamte erworbene Planungswissen verworfen werden muss. Die Planungszeiten sind so wesentlich kürzer als bei einer kompletten Neuplanung. Für die Reaktive Steuerung wird in [71] ein Zwischenzielpunkt durch die Schnittpunktberechnung des generierten Wegeplanes mit dem Rand des Sensorauffassbereiches vorgegeben. Befindet sich dieser Schnittpunkt im Inneren eines Hindernisses, wird er in den nächsten hindernisfreien Sektor verlegt.

Zur Koordinierung von Multirobotersystemen für eine Formationsfahrt wird in [5] ein *führender* Roboter definiert, dem alle anderen Roboter unter Einhaltung ihrer Formationsposition folgen. Diese einzuhaltenden Positionen werden als anzufahrende Zielpunkte für die einzelnen Roboter vorgegeben. Soll die Robotergruppe durch enge Passagen geführt werden, ordnen sich die Roboter hintereinander an, wobei der anzufahrende Zielpunkt eines Roboters die aktuelle Position seines „Vordermanns“ ist. Um dabei einen Mindestabstand zum vorderen Roboter einzuhalten, werden die entsprechenden Steuerkommandos proportional zum Abstand begrenzt. Für das Kollisionsvermeidungssystem des Miniroboters Khepera wird in [134] eine hierarchisch aufgebaute Steuerung aus zwei Fuzzy-Systemen verwendet. Das übergeordnete Fuzzy-System ist für die Kollisionsvermeidung verantwortlich, wobei es anhand der Entfernungssensoren und des körperfesten Peilungswinkels zum Zielpunkt einen körperfesten Steuerkurs und einen Geschwindigkeitsfaktor generiert. Das unterlagerte Fuzzy-System dient zur Zielpunktverfolgung. Aus seinen Eingangsdaten, dem Peilungswinkel und dem Abstand zum vorgegebenen Zielpunkt, ermittelt es die Drehgeschwindigkeiten für das linke und rechte Antriebsrad, die mit dem Geschwindigkeitsfaktor multipliziert werden. Da das übergeordnete Fuzzy-System als Ausgangsgröße nur den Winkel bereitstellt, berechnet ein internes Steuerprogramm einen virtuellen Zielpunkt, der in der vorgegebenen Winkelrichtung und im konstanten Abstand vor dem Roboter liegt. Von dieser Position werden dann die Eingangsgrößen für das unterlagerte Fuzzy-System bestimmt. Werden von den Entfernungssensoren keine Hindernisse mehr detektiert, führt das Steuerprogramm den virtuellen Zielpunkt auf den realen Zielpunkt zurück.



## 7.2 Konzept der Zielpunktgenerierung

### 7.2.1 Anforderungen

Das Submodul *Zielpunktgenerierung* berechnet einen Rendezvouspunkt mit dem aktuellen Missionsplan beim Empfang neuer Objekt- oder Manöverdaten. Eine zyklische Abarbeitung in jeder Updateroutine des Moduls *FIS*, wie sie bei den anderen Submodulen erfolgt, ist daher beim Submodul *Zielpunktgenerierung* nicht notwendig. Die Anforderungen hinsichtlich der maximalen Objekt- und Manöveranzahl entsprechen denen des Submoduls *Kollisionsüberwachung*, die in Abschnitt 2.2.1 vorgestellt wurden. Dabei sind maximal 50 Objekte und 20 Basismanöver zu bearbeiten. Durch den berechneten Rendezvous- bzw. Zielpunkt soll das Fahrzeug nach einer Objektidentifikation oder einem Ausweichmanöver auf den Missionsplan zurückgeführt werden. Dabei gibt es für die Zielpunktpositionierung zwei Forderungen:

1. Es sollen möglichst wenig Verluste an durchzuführenden Missionsaufgaben infolge nicht befahrener Manöverelemente entstehen.
2. Unter Beachtung der Manövriereigenschaften des Fahrzeuges soll ein kollisionsfreies Einfahren auf den Missionsplan möglich sein.

Die beiden Forderungen stehen im Widerspruch zueinander, wobei die letzte Forderung die höhere Priorität besitzt und auf jeden Fall einzuhalten ist.

### 7.2.2 Vorstellung des Verfahrens

Nachfolgend wird das neu entworfene Verfahren zur Bestimmung des Zielpunktes im Überblick vorgestellt. Die Idee des Verfahrens basiert auf einer Kollisionserkennung und setzt sich aus zwei Phasen zusammen (siehe Abbildung 7.2 und Abbildung 2.2). In der ersten Phase wird die Liste des aktuellen und der  $m-1$  zukünftigen Manöver von der Endposition des letzten Manövers aus rückwärts „durchfahren“. Die erste ermittelte „Kollisionsposition“ mit dem Beobachtungsbereich eines Objektes (siehe Abschnitt 1.3.2) entspricht dem anzufahrenden Zielpunkt  $x_{Ziel}$ . Da durch die Lage der Objekte und die gewählte Größe ihrer Beobachtungsbereiche nicht immer sichergestellt werden kann, dass der berechnete Zielpunkt ein kollisionsfreies Einfahren auf dem Missionsplan ermöglicht, wird dies in einer zweiten Phase überprüft.

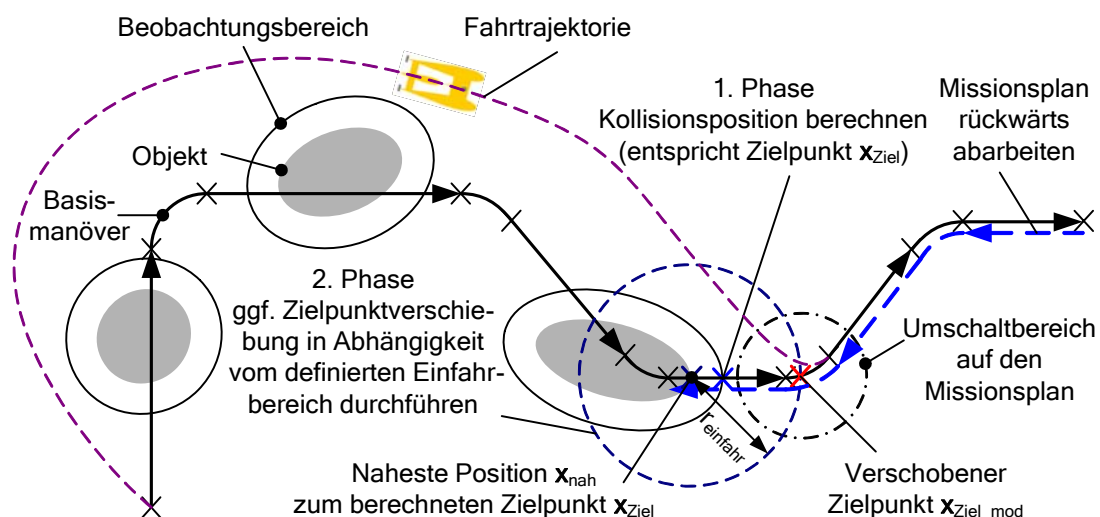


Abbildung 7.2: Bestimmung des Zielpunktes

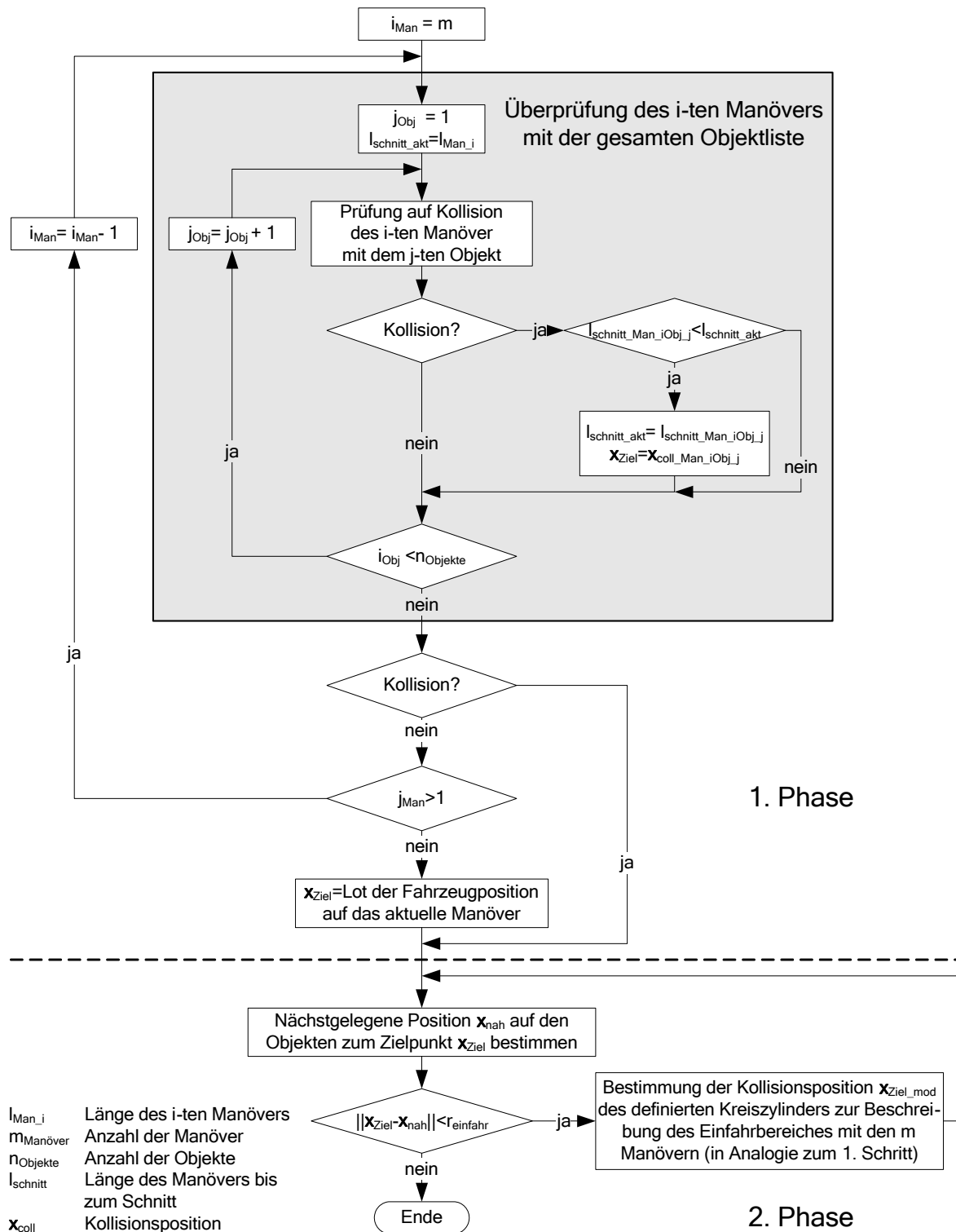


Abbildung 7.3: Flussdiagramm zur Bestimmung des Zielpunktes

Dazu wird die nächstgelegene Position  $x_{nah}$  auf den Objekten zum berechneten Zielpunkt  $x_{Ziel}$  bestimmt. Unterschreitet der Abstand zwischen den Positionen  $x_{nah}$  und  $x_{Ziel}$  einen vorgegebenen Wert  $r_{einfahr}$ , wird um die Position  $x_{nah}$  ein Kreiszylinder definiert, der eine verbotene Zone beschreibt. In dieser Zone darf nicht auf den Missionsplan eingefahren werden. Durch die Wahl des Radius  $r_{einfahr}$  der zylindrischen Grundfläche wird direkt Einfluss auf das Einfahrverhalten des Fahrzeuges genommen. So kann der Radius an die aktuellen Bedingungen, z. B. einen fahrgeschwindigkeitsabhängigen Wenderadius oder eine vorhan-

dene Seeströmung, angepasst werden. Die vertikale Ausdehnung des Kreiszylinders reicht von der Meeresoberfläche bis zur maximalen Tauchtiefe. Anschließend wird die Kollisionsposition der Manöverliste mit dem definierten Kreiszylinder, der dabei als Objekt behandelt wird, berechnet. Die einzelnen Schritte der Kollisionserkennung erfolgen in Analogie zur ersten Phase. Die so berechnete Kollisionsposition entspricht dem anzufahrenden Zielpunkt  $x_{Ziel\_mod}$ . Um zu überprüfen, ob der vorgegebene Mindestabstand  $r_{einfahr}$  zu den Objekten nach dieser Zielpunktverschiebung eingehalten wird, erfolgt eine Wiederholung der zweiten Phase mit dem Zielpunkt  $x_{Ziel\_mod}$ . Dieser iterative Prozess mit der Überprüfung des Mindestabstandes und ggf. einer Zielpunktverschiebung wird solange durchgeführt, bis ein vorgegebener Mindestabstand zu den Objekten eingehalten wird. Unterschreitet das Fahrzeug einen definierten Abstand zum berechneten Zielpunkt, wird die Führung des Moduls *FIS* an die *Missionsplanabarbeitung* übergeben, wodurch das Fahrzeug auf das aktuelle Basismanöver geführt wird.

Die zur Berechnung der Kollisionssituationen verwendeten Algorithmen entsprechen denen des Submoduls *Kollisionsüberwachung* (Kapitel 2). Durch die gewählte Anzahl  $m$  an zu überprüfenden Manövern kann zum einen direkt Einfluss auf den Erfüllungsgrad der Missionsaufgaben und zum anderen auf das sichere Umfahren beim Vorhandensein mehrerer Objekte genommen werden. Liegen z. B. zwei Objekte räumlich versetzt auf dem Missionsplan, beginnt vor dem ersten Manöver der Ausweichvorgang, der bei entsprechender Wahl der Manöveranzahl erst nach dem zweiten Objekt abgeschlossen ist. Wird gleich zu Beginn, also am Ende des Missionsabschnittes der zu überprüfenden Manöverliste eine Kollision ermittelt, wird das Fahrzeug zu diesem Punkt geführt. Die frei befahrbaren Manöverabschnitte bis zu diesem Zielpunkt werden nicht mehr abgefahren, wodurch ihre Missionsaufgaben nicht erfüllt werden können. Das zwischenzeitliche Einfahren auf den Missionsplan führt zur Erhöhung der Wahrscheinlichkeit einer Kollisionssituation und setzt einen gewissen Abstand zwischen den Objekten voraus. Durch das bei den Versuchsträgern verwendete Sonarsystem mit seiner begrenzten Reichweite und durch die Manövrierfähigkeit der Fahrzeuge sind diese Betrachtungen jedoch nur von theoretischer Natur. An dieser Stelle sei angemerkt, dass bei der Missionsaufgabe *Identifikation von Objekten* alle Objekte identifiziert werden, die auf dem Missionsplan zwischen der aktuellen Fahrzeugposition und dem Zielpunkt liegen oder bei der Rückfahrt auf dem Missionsplan die Route des Fahrzeuges tangieren. Bei dieser Aufgabe entscheidet die Wahl des Zielpunktes nicht über den Erfüllungsgrad der Missionsaufgaben.

## 7.3 Bemerkungen zum Laufzeitverhalten

Auf eine ausführliche Untersuchung des Laufzeitverhaltens der beschriebenen Algorithmen zur Zielpunktgenerierung soll in diesem Kapitel verzichtet werden, da sie auch Bestandteile der Submodule *Kollisionsüberwachung* und *Reaktive Steuerung* sind und in diesem Zusammenhang bereits detailliert untersucht wurden. An dieser Stelle erfolgt lediglich eine Abschätzung der notwendigen Rechenzeit aus den gesammelten Ergebnissen. Die Gesamtrechenzeit setzt sich aus zwei Teilen zusammen. Der erste Teil berücksichtigt die Abarbeitungszeit der ersten Phase. Geht man bei den Testbedingungen von 50 Objekten und 20 Basismanövern aus, beträgt die Rechenzeit nach Abschnitt 2.4.4 0,7 ms. Die Rechenzeit in der zweiten Phase setzt sich aus der Ermittlung der nächstgelegenen Position auf den Objekten zum berechneten Zielpunkt und der Bestimmung der Kollisionssituation mit den  $m$

Manövern und einem Objekt (Kreiszyylinder) zusammen. Die Ermittlung der kürzesten Entfernung wird bei der *Reaktiven Steuerung* durchgeführt und beträgt nach Abschnitt 5.4.3 0,42 ms. Die Zeit für die anschließende Kollisionsberechnung bestimmt sich aus der anteilmäßigen Rechenzeit für ein Objekt und allen Manövern zu 0,0146 ms ( $1/48 \cdot 0,7$  ms). Die Rechenzeit der zweiten Phase beträgt damit 0,4346 ms. Bei den untersuchten Testszenarien hat sich gezeigt, dass nach maximal zwei Wiederholungen der zweiten Phase die Zielpunktverschiebung abgeschlossen ist. So beträgt die Gesamtrechenzeit der Zielpunktgenerierung für das vorgegebene Testszenario 1,289 ms ( $0,42 \text{ ms} + 2 \cdot 0,4346 \text{ ms}$ ).

## 7.4 Zusammenfassung

In diesem Kapitel wurden die entworfenen und simulativ getesteten Algorithmen zur Zielpunktgenerierung vorgestellt. Die durchgeführte Literaturrecherche ergab wenig praktikable Ansätze. In vielen Arbeiten zur Kollisionsvermeidung wird der Zielpunkt als gegeben vorausgesetzt. Die Betrachtung einer vollautomatischen Mission mit Abfahren eines Missionsplanes und selbstständigem Ausweichen bei vorhandenen Hindernissen wird in den zugänglichen Arbeiten im Bereich der Autonomen Systeme nicht behandelt. Die Bestimmung eines Zielpunktes ist jedoch notwendig, um nach einer Aktivierung des Moduls *FIS* zur Erfüllung einer Sondersituation das Fahrzeug wieder auf den Missionsplan zurückzuführen. Der berechnete Zielpunkt soll auf der Bahn des Missionsplanes liegen und dabei zu möglichst wenig Verlusten an durchzuführenden Missionsaufgaben durch nicht befahrene Manöverelemente führen und ein kollisionsfreies Einfahren auf die Sollbahn ermöglichen. Um dies zu erreichen, erfolgt mit den Basismanövern des Missionsplanes eine Kollisionserkennung für die Objekte. Dabei werden die Basismanöver in umgekehrter Reihenfolge von den Endpositionen beginnend abgearbeitet, bis eine Kollisionssituation und damit ein Zielpunkt ermittelt wurde. Um das kollisionsfreie Einfahren zu überprüfen, wird der kürzeste Abstand vom Zielpunkt zu den umgebenden Objekten bestimmt. Unterschreitet dieser Abstand einen vorgegebenen Wert, findet eine Zielpunktverschiebung statt. Dieses Vorgehen hat sich für die untersuchten Szenarien als sehr robust und erfolgreich erwiesen.

## 8 Führungsalgorithmen

In diesem Kapitel werden die entwickelten *Führungsalgorithmen* zur Steuerung des Autonomen Unterwasserfahrzeuges in Sondersituationen vorgestellt. Die Aufgabe der Führungsalgorithmen ist es, die von den Submodulen *Kollisionsvermeidungssystem* oder *Identifikationssystem* generierten Führungsvorgaben für das Fahrzeug unter Verwendung des externen *Autopiloten* zu erfüllen. Dazu sind für die einzelnen Führungsaufgaben separate Regler implementiert, deren Stellgrößen die Vorgaben für den Autopiloten in Form der Sollgeschwindigkeit, der Solltiefe und des Sollkurses bilden. Diese universelle Schnittstelle zu den Autopiloten erlaubt die Verwendung der Algorithmen des Moduls *FIS* auch in anderen AUVs, unabhängig von ihrer vorhandenen Aktuatorkonfiguration.

Der einführende Teil dieses Kapitels stellt einen kurzen Überblick bekannter Ansätze zur Führung von mobilen Systemen hinsichtlich der verwendeten Regelungsstruktur und der Regelkonzepte vor. Ausgehend von einer regelungstechnischen Analyse, welche die vorgegebene Regelungshierarchie und das nichtlineare Systemverhalten des Unterwasserfahrzeuges vorstellt, erfolgt die Synthese der einzelnen Regelungen sowie eine Beschreibung zur Reglerparameterbestimmung auf Basis eines rechnergestützten nichtlinearen Reglerentwurfes.

### 8.1 Bekannte Ansätze

Zur Führung eines mobilen Systems sind robuste und zuverlässige Algorithmen zur Bewegungssteuerung eine primäre Voraussetzung. Solche Algorithmen steuern die Bewegungszustände des Fahrzeuges in Abhängigkeit von der geforderten Führungsaufgabe. Die dabei zu lösenden Regelungsaufgaben sind für land-, luft- und wassergeführte mobile Systeme ähnlich. Eine häufige Aufgabe besteht neben den Basisfunktionalitäten der Fahrzeugausrichtung und der Geschwindigkeitssteuerung im exakten Folgen einer vorgegebenen Trajektorie.

In [71] erfolgt die Bewegungsführung eines mobilen Roboters in einem hierarchischen Regelungskonzept, das als Drei-Ebenen-Architektur aufgebaut ist. In der obersten Ebene wird eine Fahrtroute auf Basis des globalen Wissens des Operationsgebietes generiert. Dieser Route folgt die mittlere Ebene „Lokale Kollisionsvermeidung“ unter Verwendung der aktuellen Sensordaten. Sie erzeugt eine kollisionsfreie Bewegungsreferenz für die untere Ebene „Robuste Bewegungsregelung“. Um die Parameterunsicherheiten im dynamischen Modell beim Entwurf der Regelung zu kompensieren, wurden in dieser Arbeit Gleitregler (engl.: sliding mode control) zum Folgen der Gradientenlinien eingesetzt. Gleitregler besitzen eine hohe Robustheit gegenüber ungenauen Modellparametern, was jedoch zu Lasten des Stellaufwandes geht [71]. Solche Regler haben ein diskontinuierliches Regelungsgesetz, welches im Idealfall mit unendlicher Frequenz umschalten kann. Die Gradientenfolgeregelung basiert auf der Ausrichtung des Geschwindigkeitsvektors des Roboters zum Gradientenvektor. Die Fahrgeschwindigkeit wird dabei der Krümmung des Gradienten angepasst.

Moderne Flugregelungssysteme besitzen einen hierarchischen Aufbau, bestehend aus mehreren Ebenen bzw. Regelschleifen. Ein solches System ist dadurch gekennzeichnet, dass es möglichst einfach, klar und überschaubar ist und dies auch noch nach mehrfacher Erweiterung bleibt. Das Sicherheitsprinzip einer solchen Systemstruktur besteht darin, dass Fehler in einer oberen Ebene keine Auswirkungen auf die unteren Ebenen haben dürfen, und dass trotz Ausfalls einer oberen Ebene die unterlagerten Ebenen weiterhin funktionieren [27]. In der untersten Ebene befinden sich die Regler („Dämpfer“) zur Erhöhung der Eigendämpfung und Störunterdrückung. In der darüber liegenden Ebene sorgen die „Basisregler“ für Stabilisierung der Fluglage. Die nächste Ebene umfasst die Regler zur Bahnführung. Der Entwurf der einzelnen Regelschleifen erfolgt dabei schrittweise von „innen“ nach „außen“ [27]. Durch das nichtlineare Systemverhalten von Flugkörpern erfolgt die Auslegung im Allgemeinen durch einen linearen Regelungsentwurf in den relevanten Arbeitspunkten. Als Regler werden neben den klassischen PID-Reglern auch Zustandsregler eingesetzt. Die Reglerstrukturen sind meist dadurch gekennzeichnet, dass sie neben den Regleralgorithmen auch Filter, Kennlinien, Begrenzer, Vergleichoperatoren und Funktionen enthalten, um das gewünschte Regelungsverhalten zu erzielen. Die arbeitspunktabhängigen Regelungsparameter sind vielfach in Lookup-Tabellen hinterlegt und werden in Abhängigkeit von der aktuellen Flugsituation (Staudruck, statischer Druck, Machzahl) an den Regler übergeben. Eine solche Anpassung der Reglerparameter wird heute standardmäßig in allen Flugreglern eingesetzt [27]. In [100] erfolgt diese Adaption durch Fuzzy-Logik, wodurch eine variable Nutzung der Flugzustandsgrößen zur Beschreibung der aktuellen Flugsituation möglich ist. Dadurch können in bestimmten Flugphasen mehr Zustandsgrößen als in anderen eingesetzt werden, was zum einen zu einer genaueren Beschreibung der Flugsituation und zum anderen zu einer Senkung der verwendeten Parameter führt.

Die häufigsten Regelungs- und Führungsaufgaben bei Schiffen und Unterwasserfahrzeugen umfassen neben der Basisfunktionalität der Lagestabilisierung, die Bahnverfolgung und die Positionierung. Die Regelungsstrukturen sind sowohl als mehrstufige Kaskaden aber auch als einschleifige Regelkreise aufgebaut. In [67] sind die Regelungs- und Führungsaufgaben als einzelne Fuzzy-Systeme in einer Mehrebenen-Struktur implementiert. Ein Regelungskonzept für das AUV AQUA EXPLORER 1000 zur Kabelverfolgung und Höhenregelung (das AUV hält einen vorgegebenen Sollabstand zum Meeresboden ein) auf Basis von PID-Reglern wird in [7] vorgestellt. Die Reglerparameter werden in Abhängigkeit von der Fahrgeschwindigkeit verändert, um das geschwindigkeitsabhängige hydrodynamische Verhalten zu berücksichtigen. Einen kaskadenartigen Aufbau der Regelungen gibt es jedoch nicht, d.h. die Stellgrößen der Führungsregelungen sind die Sollaktuatorwerte des AUVs. Für die Wandverfolgung des Versuchsträgers AUV „Tri-Dog 1“ werden in [95] P-PI Kaskadenstrukturen zur Entfernung- und Peilungswinkelregelung verwendet. Die inneren Regelkreise regeln die jeweiligen Ableitungen der Führungsgrößen. In [58] wird eine Bahnregelung für ein Schiff auf Basis eines LQG-Reglers (linear quadratischer gausscher Regler) beschrieben. Dieser Reglertyp findet Anwendung, wenn gestörte Ausgangsgrößenmessungen vorliegen und nicht alle Systemzustände messbar sind. Der Regler verwendet ein erdfestes Koordinatensystem, um die gestörte GPS-Position als Zustandsgröße nutzen zu können. Als weitere Zustandsgrößen werden der gestörte Gierwinkel der Kreismessung, die Werte zur Beschreibung des Wellenmodells, die körperfesten Geschwindigkeiten und Winkelgeschwindigkeiten sowie die Parameter zur Beschreibung des Modellfehlers und der Umweltstörungen verwendet.

## 8.2 Regelungstechnische Analyse

In diesem Abschnitt erfolgt eine Beschreibung der entwickelten Regelungshierarchie. Dabei wird die zur Verfügung stehende Schnittstelle zum Autopiloten erläutert und auf einige Probleme bei der Modellierung des geregelten Fahrzeugverhaltens eingegangen. Durch die Vorstellung des physikalischen Fahrzeugmodells erhält der Leser ein besseres Verständnis über die vorhandenen Nichtlinearitäten und Steuermöglichkeiten eines Unterwasserfahrzeuges. Es stellt die Grundlage für den optimalen modellgestützten Entwurf der Regelungen dar.

### 8.2.1 Regelungshierarchie

Die in dieser Arbeit als Versuchsträger eingesetzten AUVs verwenden eine kaskadenartige Regelungshierarchie, wie sie in Abbildung 8.1 dargestellt ist. Dabei liegt die innerste Regelschleife im *Autopiloten* und dient zur Ausregelung der Fahrzeuglage und -geschwindigkeit. In der darüber liegenden Ebene befinden sich die Regelungen zur Führung des Fahrzeuges. Diese Ebene ist in Abhängigkeit vom verwendeten, fahrzeugspezifischen Autopiloten unterschiedlich stark ausprägt. Obligatorisch befindet sich dort eine Tiefenregelung. Zusätzlich können sich in dieser Ebene die Regler zur Positionierung, Kurs- und Bahnführung des Fahrzeuges befinden. Deshalb werden alle nicht obligatorisch im Autopiloten bearbeiteten Führungsaufgaben im Modul *FIS* durchgeführt. So besitzt der *Autopilot* als Führungsgrößen den Kurs, die Tiefe und die Geschwindigkeit. Beim Kurs unterscheidet man zwei Betriebsarten, *Kurs über Grund* und *Gierwinkel* (siehe Abschnitt 5.4.2). Für die Tiefenregelung des Autopiloten kann eine *Tiefe* oder eine *Höhe über Grund* (siehe Abschnitt 3.2.4) vorgegeben werden.

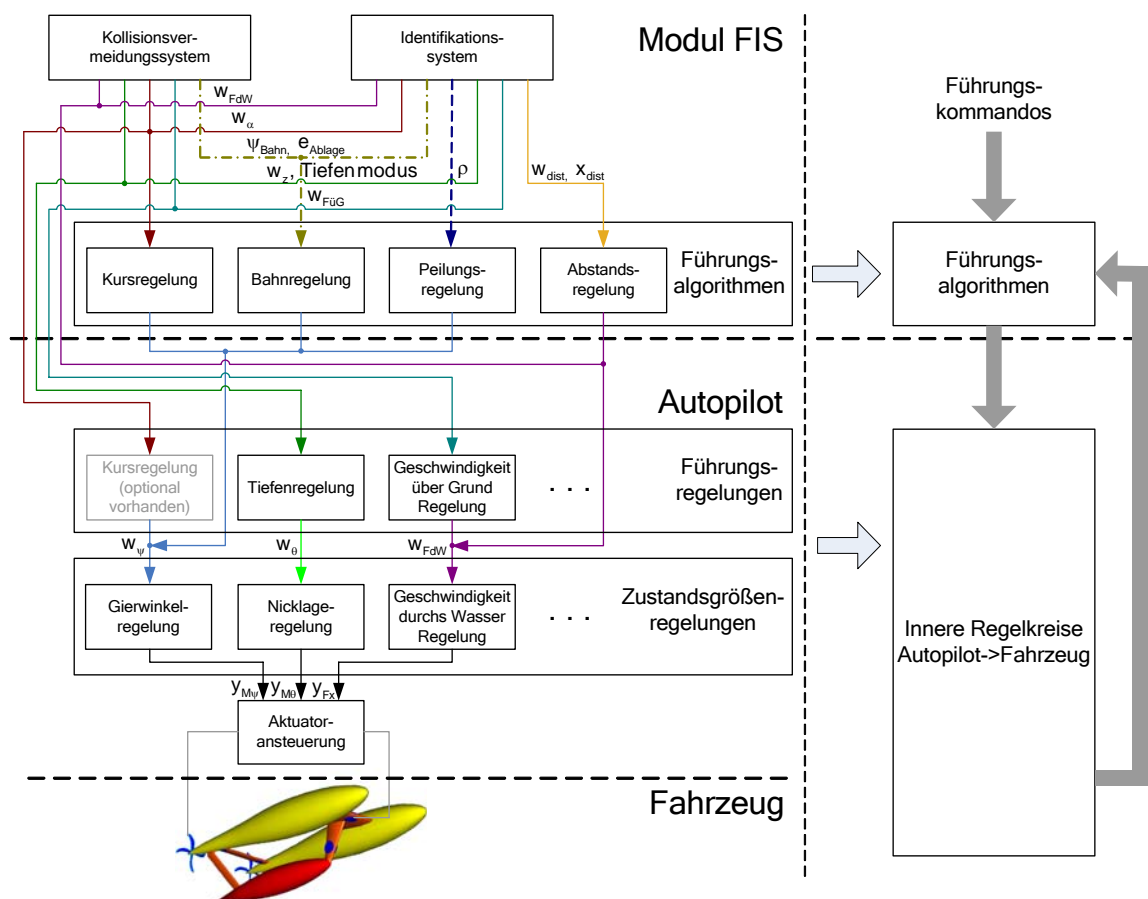


Abbildung 8.1: Regelungshierarchie

Als Geschwindigkeit wird die *Geschwindigkeit durchs Wasser* (d. h. die Geschwindigkeit des Fahrzeuges gegenüber dem umgebenden Wasser) oder die *Geschwindigkeit über Grund* (Geschwindigkeit gegenüber einem erdfesten Bezugspunkt) durch den *Autopiloten* geregelt. Nur über diese Schnittstelle kann das Modul *FIS* auf das Fahrzeugverhalten Einfluss nehmen.

Zur Durchführung der einzelnen Reglerentwürfe des Submoduls *Führungsalgorithmen* ist es durch die vorgegebene Regelungshierarchie notwendig, dass die Regelungen des *Autopiloten* voll funktionstüchtig sind und ihr Systemverhalten bekannt ist. Da zu Projektbeginn noch kein Fahrzeug bzw. Fahrzeugmodell zur Verfügung stand und damit auch kein Autopilot, wurde das Systemverhalten des mit dem *Autopiloten* geregelten Fahrzeuges (innere Regelkreise) durch Kennlinien, P-T1 und P-T2 Glieder approximiert [103]. Das nachzubildende Systemverhalten wurde aus der Systemspezifikation des Projektes „DeepC“ (Maximalwerte der Geschwindigkeiten, der Drehgeschwindigkeiten, der Fahrzeuglagen und der Antriebsleistungen) und den ersten hydrodynamischen Berechnungen/Abschätzungen der Firma ATLAS ELEKTRONIK GmbH entnommen. Bei den späteren Versuchen an der DeepC-Testumgebung, unter Verwendung des entwickelten *Autopiloten* und eines entworfenen Fahrzeugmodells, stellten sich Abweichungen zum einfachen Regelkreis-Modell hinsichtlich der angenommenen Dynamik und des Regelverhaltens heraus. Es zeigte sich, dass das Regelverhalten stark arbeitspunktabhängig ist, was auch nicht durch die Adaption der Regelungsparameter kompensiert werden konnte. Ein Grund liegt in der geschwindigkeitsabhängigen Verwendung der Aktuatoren und einer damit verbundenen Änderung des Fahrzeugverhaltens (siehe Abschnitt 1.1.1). Eine Anpassung bzw. Erweiterung des Modells des geregelten Fahrzeugverhaltens unter Verwendung der Simulationsauswertungen ist sehr aufwendig und durch die zur Verfügung stehenden Algorithmen des Autopiloten und des Fahrzeuges in Verbindung mit dem verwendeten Verfahren der Reglerparameterbestimmung (siehe Abschnitt 8.4) auch nicht notwendig. Bei diesem Verfahren werden die Reglerparameter durch eine Optimierung ermittelt, wobei die Gütewerte durch die Auswertung des Führungsverhaltens der simulierten Regelkreise gebildet werden. Eine analytische Beschreibung der Regelkreise (z.B. durch Übertragungsfunktionen) ist für dieses Entwurfsverfahren nicht notwendig. Deshalb wurde in einem zweiten Entwurf sowohl das Fahrzeugmodell als auch der entwickelte Autopilot in die Simulation eingebunden. Dazu wurde unter Simulink [142] der Programmcode des Fahrzeugmodells und des Autopiloten in S-Funktionen implementiert (siehe auch Abschnitt 9.1). Veränderungen bzw. Erweiterungen im Autopiloten oder im Fahrzeugmodell können dann direkt in die einzelnen Funktionen übernommen werden, ohne dass eine Nachbildung des geänderten Führungsverhaltens im o. g. Regelkreis-Modell durchgeführt werden muss. Nachfolgend soll das eingesetzte Fahrzeugmodell kurz vorgestellt werden. Eine Beschreibung der Regelungsalgorithmen des *Autopiloten* erfolgt in dieser Arbeit nicht, da es für die einzelnen Fahrzeuge unterschiedliche Konzepte gibt und es zum Verständnis des Regelungsentwurfes auch nicht notwendig ist.

## 8.2.2 Modellierung des Fahrzeuges

Für die Durchführung des Reglerentwurfes und zur simulativen Erprobung der einzelnen Module in der DeepC-Testumgebung ist die Erstellung eines Systemmodells des Fahrzeuges erforderlich. Die Beschreibung des Systemverhaltens von Unterwasserfahrzeugen basiert vielfach auf einem physikalischen Modell [48, 58]. Die physikalischen Modelle erfordern gegenüber den durch experimentelle Prozessanalyse entwickelten Modellen einen Mehrauf-



wand bei der Erstellung der Modellstruktur und der Parameter [49]. Sie können dann aber auch die wahren Zusammenhänge und Wechselwirkungen innerhalb des Systems und mit der Umwelt exakter nachbilden. Einzelne, bereits bekannte Parameter (z.B. Hebellängen, Massen, Ruderflächen, Propellerkennwerte) können direkt in das Modell übernommen werden. Grundlage des physikalischen Modells des Unterwasserfahrzeuges ist eine nichtlineare Differenzialgleichung, die alle am Fahrzeug wirkenden Kräfte und Momente beinhaltet und zur Bestimmung der Zustandsgrößen (Position, Winkel, Geschwindigkeit, Winkelgeschwindigkeit) eingesetzt wird [58, 59].

Durch die auf das Fahrzeug wirkenden Kräfte und Momente  $\tau_{RB}$  entstehen nach dem 2. Newtonschen Axiom translatorische und rotatorische Bewegungen am Fahrzeug. Sie haben ihre Ursachen in der Trägheits-, Coriolis- und Zentripetalkraft und können durch die Gleichung:

$$\mathbf{M}_{RB} \dot{\mathbf{v}} + \mathbf{C}_{RB}(\mathbf{v}) \mathbf{v} = \tau_{RB} \quad (8.1)$$

beschrieben werden. Dabei umfasst der Vektor  $\mathbf{v}$  die Geschwindigkeiten  $\mathbf{v}_1 = [u, v, w]^T$  und Drehgeschwindigkeiten  $\mathbf{v}_2 = [p, q, r]^T$  im fahrzeugfesten Koordinatensystem.  $\mathbf{M}_{RB}$  enthält die Trägheitsmatrix. Die Terme zur Beschreibung der Coriolis- und Zentripetalkräfte bzw. -momente sind in der Matrix  $\mathbf{C}_{RB}$  enthalten.

Die externen Kräfte und Momente, welche auf das Fahrzeug wirken, werden durch die Gleichung:

$$\tau_{RB} = \underbrace{\tau_{Antrieb}}_{\text{resultierende Antriebskräfte}} - \underbrace{\mathbf{M}_A \cdot \dot{\mathbf{v}} - \mathbf{C}_A(\mathbf{v}) \mathbf{v}}_{\text{hydrodynamische Zusatzmassen}} - \underbrace{\mathbf{D}(\mathbf{v}) \mathbf{v}}_{\text{hydrodynamische Widerstände und Auftrieb}} - \underbrace{\mathbf{g}(\boldsymbol{\eta})}_{\text{Auftrieb infolge der Gravitation}} \quad (8.2)$$

beschrieben. Der Term der hydrodynamischen Zusatzmassen charakterisiert die zu verrichtende Arbeit, die das Fahrzeug bei einer Beschleunigung oder Verzögerung leistet, um die es umgebende Flüssigkeit zu bewegen. Durch Zusammenfassung der Gleichungen (8.1) und (8.2) entsteht die nichtlineare Bewegungsgleichung des Fahrzeuges:

$$\begin{aligned} \mathbf{M} \cdot \dot{\mathbf{v}} + \mathbf{C}(\mathbf{v}) \mathbf{v} + \mathbf{D}(\mathbf{v}) \mathbf{v} + \mathbf{g}(\boldsymbol{\eta}) &= \tau_{Antrieb} \\ \mathbf{M} &\text{ Trägheitsmatrix (incl. Zusatzmassen) mit} \\ &\mathbf{M} = \mathbf{M}_{RB} + \mathbf{M}_A \\ \mathbf{C}(\mathbf{v}) &\text{ Matrix der Coriolis- und Zentripetaleinflüsse} \\ &\text{(incl. Zusatzmassen) } \mathbf{C}(\mathbf{v}) = \mathbf{C}_{RB}(\mathbf{v}) + \mathbf{C}_A(\mathbf{v}) \\ \mathbf{D}(\mathbf{v}) &\text{ hydrodynamische Widerstands- und} \\ &\text{Auftriebsmatrix} \\ \mathbf{g}(\boldsymbol{\eta}) &\text{ Vektor der resultierenden Auftriebskräfte} \\ \tau_{Antrieb} &\text{ Antriebskräfte.} \end{aligned} \quad (8.3)$$

Die Kräfte und Momente, welche durch den hydrodynamischen Widerstand und Auftrieb verursacht werden, können komponentenweise nach ihrer Entstehung an den einzelnen Fahrzeugmodulen und der dort wirkenden Anströmung berechnet und additiv zu  $\mathbf{D}(\mathbf{v}) \mathbf{v}$  zusammengefasst werden. Die Bestimmung der Kräfte und Momente in der Beziehung  $\mathbf{D}(\mathbf{v}) \mathbf{v}$  erfolgt im Allgemeinen in der Form:

$$\begin{aligned}
F &= \frac{\rho}{2} \cdot v_{\text{Ström}}^2 \cdot A \cdot (C^A + C^W) \\
M &= \frac{\rho}{2} \cdot v_{\text{Ström}}^2 \cdot A \cdot L \cdot C^M \\
\rho &\quad \text{Dichte des Wassers} \\
v_{\text{Ström}} &\quad \text{Strömungsgeschwindigkeit} \\
A &\quad \text{Fläche} \\
L &\quad \text{Bezugslänge} \\
C^A &\quad \text{Auftriebsbeiwert} \\
C^W &\quad \text{Widerstandsbeiwert} \\
C^M &\quad \text{Momentenbeiwert.}
\end{aligned} \tag{8.4}$$

Die Beiwerte sind vom Anströmwinkel, der Strömungsgeschwindigkeit und von konstruktiven Gegebenheiten des Körpers abhängig. Der Funktionsverlauf der einzelnen Beiwerte wird in Abhängigkeit vom Anströmwinkel durch Experimente (Schleppversuche) oder durch theoretische Modellierung ermittelt. Diese Funktionsverläufe zeichnen sich durch eine starke Nichtlinearität aus. Eine weitere Nichtlinearität besteht durch die quadratische Beziehung der Anströmgeschwindigkeit  $v_{\text{Ström}}$  zum hydrodynamischen Widerstand und Auftrieb. Die Berechnung der resultierenden Auftriebskräfte  $\mathbf{g}(\eta)$  im körperfesten Koordinatensystem erfolgt durch Transformation der erdbezogenen Auftriebskraft  $\mathbf{B}$  und der Gewichtskraft  $\mathbf{W}$  unter Beachtung ihrer Wirkungsrichtung. Die Kräfte und Momente der Antriebseinheiten  $\tau_{\text{Antrieb}}$  werden separat für die Komponenten Propeller, Querstrahler und Ruder berechnet. Für die Ermittlung der vom Propeller erzeugten Kräfte und Momente werden die hydrodynamischen Beziehungen des Propellers verwendet. Die Algorithmen und die Parameter des auf dieser Basis erstellten Fahrzeugmodells wurden vom Projektpartner ATLAS ELEKTRONIK zur Verfügung gestellt und unter Simulink [142] implementiert.

## 8.3 Synthese der Führungsregelungen

In diesem Abschnitt werden die einzelnen, neu entwickelten Regler des Submoduls *Führungsalgorithmen* im Überblick vorgestellt. Ausgehend von den Anforderungen an die Regelungen und einer Beschreibung des verwendeten Reglertyps erfolgt eine kurze Darlegung der einzelnen Regelungsaufgaben und der dabei verwendeten Regelungsstruktur.

### 8.3.1 Anforderungen an die Regelungen

Durch das nichtlineare Systemverhalten der unterlagerten Regelkreise des *Autopiloten* und zum Erreichen des gewünschten Regelverhaltens im gesamten Arbeitsbereich, unter Berücksichtigung der vorhandenen Umweltbedingungen (Seeströmung), ist eine robuste Auslegung der Regelungen erforderlich. Sie sollen dabei ein gutes Führungs- und Störverhalten besitzen. Alle eingesetzten Regelungen arbeiten in der Horizontalebene, da die Tiefenvorgaben direkt vom Kollisionsvermeidungs- bzw. Identifikationssystem an den *Autopiloten* übergeben werden. Die in den einzelnen Regelungen verwendeten Navigationsdaten liegen in einer aufbereiteten Form vor. Zur Erfüllung dieser Anforderungen wird ein adaptiver PID-Regler eingesetzt. Die Verwendung eines Zustandsreglers ist durch die vorgegebene Regelungshierarchie, die keine direkte Aktuatoransteuerung unterstützt, nicht möglich.

### 8.3.2 Adaptiver PID-Regler

Der eingesetzte adaptive PID-Regler kann neben seiner Grundfunktionalität zusätzliche strukturelle Variationen zum Erreichen des geforderten Regelungsverhaltens besitzen. Abbildung 8.2 zeigt die Reglerstruktur des adaptiven PID-Reglers.

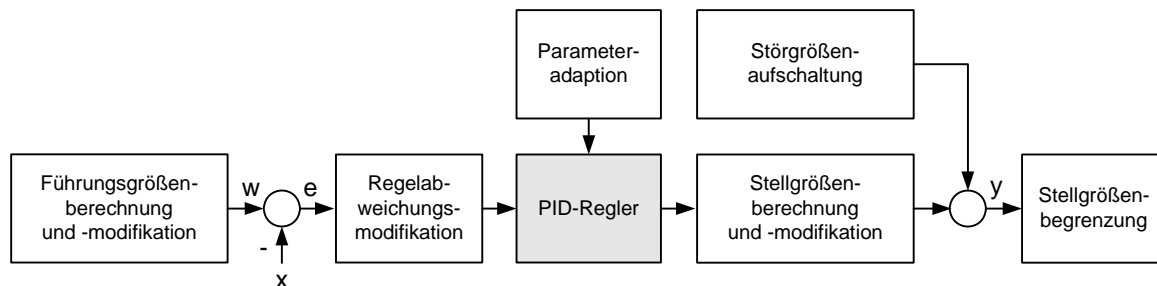


Abbildung 8.2: Reglerstruktur

Der PID-Regler verwendet einen modifizierten Algorithmus aus [85], der sich nach der folgenden Gleichung:

$$y(k) = KP \cdot e(k) + \Delta T \cdot \sum_{i=0}^{k-1} KI \cdot e(i) - KD \cdot dx(k) \quad (8.5)$$

berechnet. Gegenüber dem Basis-PID-Algorithmus wird hier die Führungsgröße  $w(k)$  nicht im Differenzierer berücksichtigt. So werden die bei schnellen Führungsgrößenänderungen auftretenden großen Stellgrößen gedämpft. Die Lage der Pole ändert sich durch diese Modifikation nicht [85]. Abbildung 8.3 zeigt das Blockschaltbild des verwendeten PID-Reglers. Der proportionale Anteil kann durch Schranken begrenzt werden. Die Begrenzung des integralen Anteils ist durch Schranken oder einen AntiWindup-Algorithmus möglich. Die einzelnen Reglerparameter  $KP$ ,  $KI$  und  $KD$  werden in Abhängigkeit des Arbeitspunktes bestimmt und an den Regler übergeben.

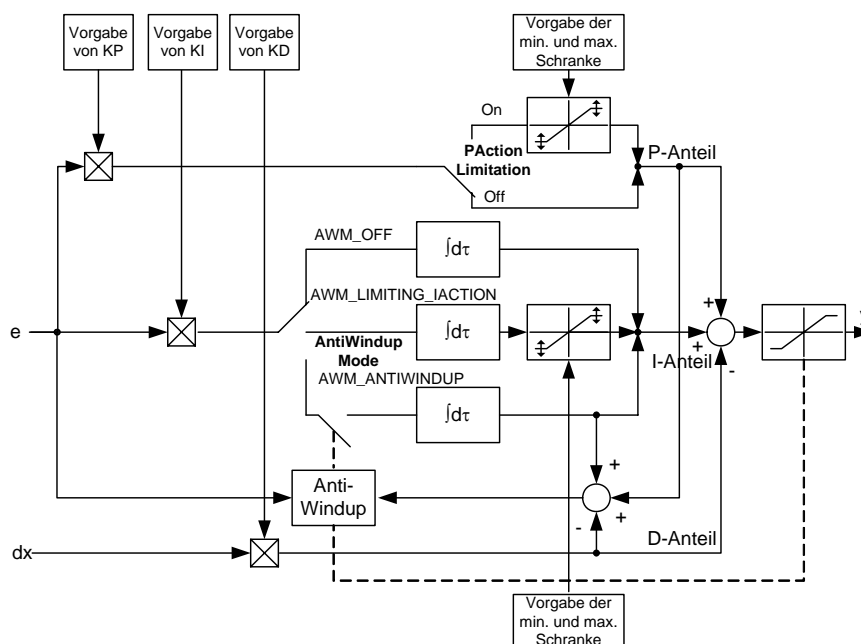


Abbildung 8.3: Blockschaltbild des PID-Reglers

### 8.3.3 Bahnregelung

Die Bahnregelung führt das Fahrzeug auf eine vorgegebene Bahn in der horizontalen Ebene. Die Stellgröße der Bahnregelung  $y_{Bahn}$  berechnet sich aus der Stellgröße eines PID-Reglers, der die horizontale Ablage  $e_{Ablage}$  ausregelt, und der aktuellen Bahnrichtung  $\psi_{Bahn}$  zu:

$$y_{Bahn} = y_{Ablage} + \psi_{Bahn} \cdot \quad (8.6)$$

Diese Stellgröße bildet den Sollwert des Gierwinkels  $w_\psi$ , der an den *Autopiloten* übergeben wird. Die Verwendung des integralen Anteils ist zum Kompensieren einer möglichen Seeströmung und einer bleibenden Regelabweichung in der Gierwinkelregelung notwendig. Die Reglerparameter werden in Abhängigkeit von der aktuellen Fahrgeschwindigkeit durchs Wasser und der Regelungsabweichung  $e_{Ablage}$  unter Verwendung von Kennlinien adaptiert. Die Bestimmung der Stützstellen dieser Kennlinien erfolgt durch eine Parameteroptimierung nach Abschnitt 8.4.1.

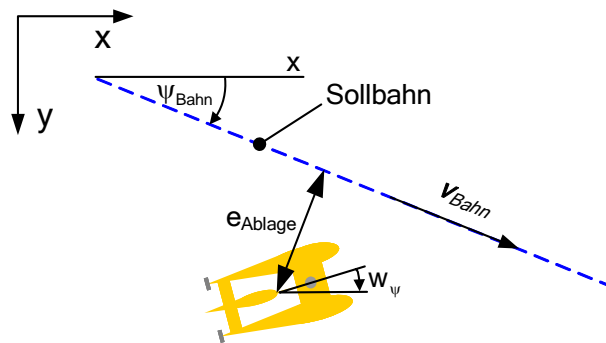


Abbildung 8.4: Parameter der Bahnregelung

### 8.3.4 Kursregelung

Da einige Autopiloten keine Ausregelung des Kurses über Grund durchführen (z.B. AUV M63), wurde diese Steuerungsaufgabe mit in die Führungsalgorithmen implementiert. Der Kursregler generiert einen Sollgierwinkel  $w_\psi$  für den *Autopiloten* durch Ausregeln der Regelabweichung  $e_{\alpha G}$  und einer Führungsgrößenaufschaltung des Sollkurses  $w_{\alpha G}$  (siehe Abbildung 8.5). Als Regler wird ein adaptiver PI-Regler eingesetzt, dessen Reglerparameter in Abhängigkeit von der Fahrgeschwindigkeit durchs Wasser adaptiert werden.

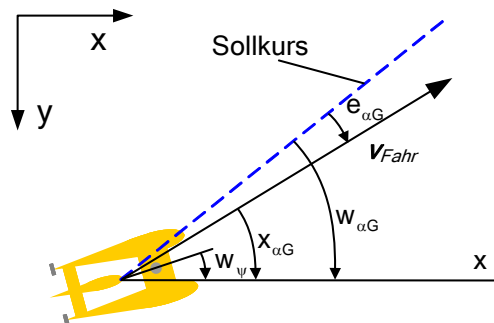


Abbildung 8.5: Parameter der Kursregelung

### 8.3.5 Abstandsregelung

Durch die Abstandsregelung soll das Fahrzeug einen vorgegebenen Vorausabstand  $w_{dist}$  zu einem Objekt einnehmen und halten. Diese Regelung ist als P-PI Kaskadenregelung aufgebaut. Eine solche Kaskadenstruktur wird üblicherweise auch bei Lageregelungen im Maschinenbau verwendet [125]. Der Führungsregler generiert abhängig von der Regelabweichung  $e_{dist}$  und der Peilungswinkelabweichung  $e_\rho$  eine Anfahrtschwindigkeit, die der PI-Folgeregler einstellt. Die Stellgröße des Folgereglers wird als Sollgeschwindigkeit durchs Wasser an den *Autopiloten* übergeben. Die Einbeziehung des Peilungswinkels bewirkt, dass bei zu großen Winkelwerten die Anfahrtschwindigkeit gedrosselt bzw. auf Null gesetzt wird. Die einzelnen Parameter der Abstandsregelung sind in Abbildung 8.6 dargestellt. Die Abstandsregelung wird immer in Kombination mit der Peilungsregelung aktiviert. Dies ist notwendig, da nur die Vorausgeschwindigkeit als Stellgröße verwendet wird und so das Fahrzeug immer auf das Objekt ausgerichtet sein muss.

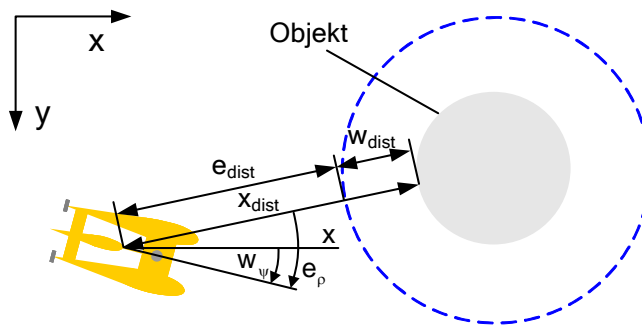


Abbildung 8.6: Parameter der Abstandsregelung

### 8.3.6 Peilungsregelung

Die Peilungsregelung richtet das Fahrzeug auf eine vorgegebene Position in der Horizontalebene aus. Als Regler wird ein adaptiver P-Regler eingesetzt, für dessen Proportionalbeiwert  $K_P$  eine Adaption in Abhängigkeit von der Fahrgeschwindigkeit und des Positionsabstandes  $x_{dist}$  erfolgt. Die Stellgröße der Peilungsregelung bildet der Sollgierwinkel  $w_\psi$  des *Autopiloten*. Er berechnet sich aus der Stellgröße des P-Reglers, der die Peilungswinkelabweichung  $e_\rho$  ausregelt, und dem aktuellen erdfesten Peilungswinkel  $\rho$  zu:

$$w_\psi = y_\rho + \rho. \quad (8.7)$$

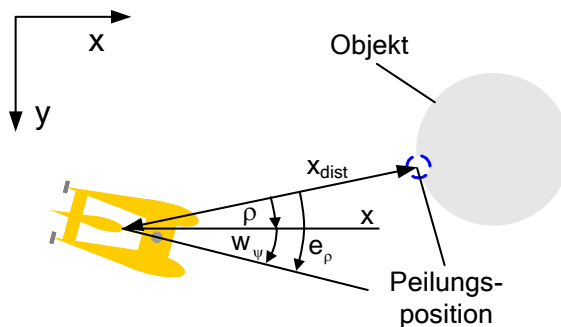


Abbildung 8.7: Parameter der Peilungsregelung

## 8.4 Nichtlinearer Regelungsentwurf

Die nachfolgenden Abschnitte beschreiben den modellgestützten, nichtlinearen Regelungsentwurf im Überblick. Der Entwurf wird im Zeitbereich unter Verwendung der FCD-Toolbox [46, 47] und des NCD-Blocksets [141] durchgeführt. Dabei werden die Reglerparameter durch das Verfahren der Parameteroptimierung mit Hilfe von Evolutionsstrategien ermittelt [50]. Durch diesen Ansatz ist es möglich, die Nichtlinearitäten der einzelnen Elemente der Regelstrecke (*Autopilot*, Fahrzeug) direkt in den Entwurf einzubeziehen. Eine systemanalytische Beschreibung der Regelstrecke und ihre Linearisierung für vorgegebene Arbeitspunkte, wie sie bei einem Entwurf im Frequenz- oder Bildbereich erforderlich ist, entfallen.

### 8.4.1 Reglerparameterbestimmung mittels Parameteroptimierung

Das Verfahren der Parameteroptimierung setzt sich aus mehreren Arbeitsphasen zusammen (siehe Abbildung 8.8). Bevor der eigentliche Optimierungsalgorithmus abgearbeitet werden kann, ist es erforderlich, die teils verbal, teils fachspezifisch vorliegenden Güteforderungen als mathematisch korrekte Gütekriterien zu formulieren. Das in dieser Arbeit verwendete Gütekriterium wird in Abschnitt 8.4.2 vorgestellt. Des Weiteren wird in dieser Phase der mögliche Suchraum durch das Definieren von Parameterbeschränkungen eingegrenzt. Dies kann so zu einer effizienteren Suche nach dem Minimum führen (bei Simplexverfahren und Evolutionsstrategien). Nach der Erstellung des Gütekriteriums und der Beschränkungen kann die eigentliche Optimierung durchgeführt werden. Dabei werden durch ein Optimierungsverfahren ein oder mehrere Reglerparameter  $x$  geändert und ihre Wirkung auf das Gütekriterium bewertet. Diese Wirkung reflektiert der Gütefunktionswert  $Q$ . Die Bestimmung des Gütefunktionswertes  $Q$  erfolgt in einem zweistufigen Prozess. In der ersten Stufe wird der modellierte Regelkreis mit den geänderten Reglerparametern rechnergestützt simuliert. Die hieraus gewonnenen Signalverläufe werden in einer zweiten Stufe zur Berechnung des Gütefunktionswertes  $Q$  benutzt. Dieser Gütefunktionswert  $Q$  wird nun an das Optimierungsverfahren übergeben und dient zur Festlegung der nächsten Parametermodifikationen mit dem Ziel, den Gütefunktionswert  $Q$  weiter zu minimieren. Der so entstandene Algorithmus läuft iterativ ab und kann erst durch die Erfüllung eines Abbruchkriteriums (z.B. maximale Anzahl an Gütefunktionsberechnungen) beendet werden.

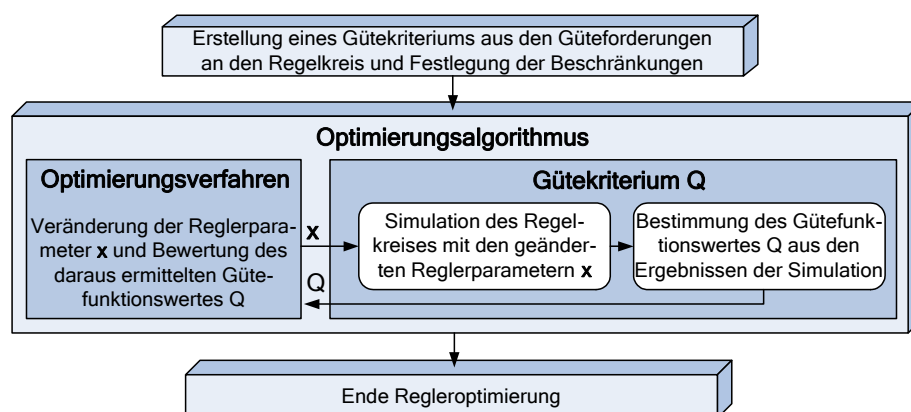


Abbildung 8.8: Ablauf der Reglerparameteroptimierung

## 8.4.2 Gütekriterien im Zeitbereich

Das gewünschte Führungsverhalten eines Regelkreises wird im Allgemeinen dadurch charakterisiert, dass die Regelgröße  $y$  der Führungsgröße  $w$  möglichst schnell und genau folgt. Als Führungsgrößensignal wird häufig die Sprungfunktion eingesetzt, da durch sie sowohl das dynamische als auch das statische Verhalten des Regelkreises untersucht werden kann. Diese Forderung kann in einer einfachen mathematischen Form in Integralkriterien umgesetzt werden. Der Inhalt dieser Kriterien beinhaltet nur verbale Güteaussagen (z.B. Minimierung der Regelabweichung, schnelles Einregeln auf den vorgegebenen Führungswert  $w$ ), durch welche ein gewünschter Führungsverlauf nur sehr ungenau beschrieben werden kann.

Eine andere Möglichkeit, das gewünschte Führungsverhalten in ein Gütekriterium zu fassen, ist die Vorgabe von Schrankenverläufen, in denen der Regelgrößenverlauf liegen soll (Abbildung 8.9). Das Verletzen der Schranken durch den Signalverlauf wird in einem Gütekriterium durch Integration ermittelt. Die nachfolgenden Gleichungen beschreiben die einzelnen Schritte zur Bestimmung des Gütwertes  $Q$ .

Schritt 1 - Bestimmung der Schrankenverletzung:

$$\begin{aligned} US(t) &= \{0 \text{ für } C_u(t) < y(t); 1 \text{ für } C_u(t) \geq y(t)\} \\ OS(t) &= \{0 \text{ für } C_o(t) > y(t); 1 \text{ für } C_o(t) \leq y(t)\} \end{aligned} \quad (8.8)$$

Schritt 2 - Gewichtete Integration der Verletzungen:

$$Q = \int_0^{t_e} (US(t) \cdot WI_u \cdot (C_u(t) - y(t)) + OS(t) \cdot WI_o \cdot (y(t) - C_o(t))) \cdot dt \quad (8.9)$$

mit

$C_u(t)$ ,  $C_o(t)$  untere und obere Schranke

$WI_u$ ,  $WI_o$  Wichtungsfaktoren.

Die Definition dieser Schranken Kriterien erfolgt in dieser Arbeit unter Verwendung des Nonlinear Control Design (NCD) Blocksets für Simulink [141]. Um bei nichtlinearen Strecken einen Regelungsentwurf durchführen zu können, wird eine Sollwertfolge über den gesamten interessierenden Arbeitsbereich definiert. So können die Gütwerte  $Q$  der einzelnen Übergänge gewichtet zu einem Gesamtkriterium  $Q_\Sigma$  zusammengefasst werden [50, 93]. Die ermittelten Reglerparameter sind so für den gesamten Arbeitsbereich optimal ausgelegt.

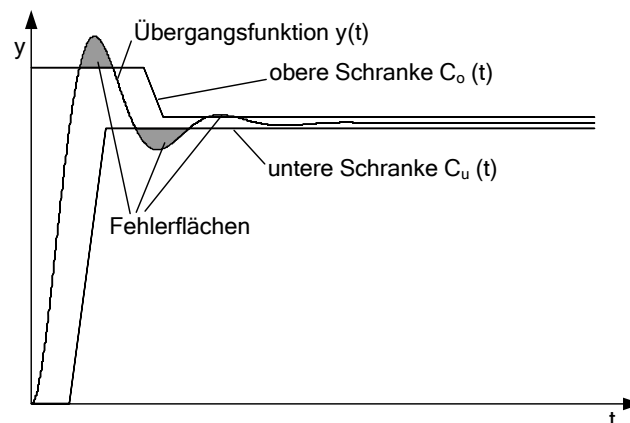


Abbildung 8.9: Schranken Kriterium

### 8.4.3 Robustheit

Eine große Praxisrelevanz besitzt die Robustheit einer Regelung. Der Robustheitsentwurf kann für die Entwurfsphase oder für den späteren Betrieb der Regelung durchgeführt werden. Eine Auslegung in der Entwurfsphase ist dann gegeben, wenn die reale Strecke nur ungenau durch ein Streckenmodell beschrieben werden kann. Hier werden Ungenauigkeitsabschätzungen vorgenommen und in den Robustheitsentwurf einbezogen. Arbeitet der Regler im Betrieb an einer Strecke mit variablen Parametern und Betriebszuständen, kann er durch einen Robustheitsentwurf für den gesamten Arbeitsbereich ausgelegt werden [48, 50]. Bei einem Unterwasserfahrzeug können variable Parameter der Strecke:

- eine variable Nutzlast,
- die Seeströmung sowie
- der Salzgehalt des Wassers sein.

Ein veränderlicher Betriebszustand ist die Fahrgeschwindigkeit durchs Wasser. Ein solches Variieren von Parametern wird als Unsicherheit bezeichnet [1]. Das Ziel eines Robustheitsentwurfes bei Regelkreisen besteht nun darin, die gewünschten Güteforderungen der Regelung, unabhängig von den möglichen Parametervariationen der unsicheren Parameter, bestmöglich zu erfüllen. Bei der in dieser Arbeit eingesetzten Software (FCD-Toolbox [47]) erfolgt der Robustheitsentwurf auf Basis der Lösung eines Multimodellproblems. Dabei werden  $l$  Betriebspunkte definiert, die jeweils eine andere Kombination der unsicheren Parameter enthalten. Diese Punkte sollten relevante Repräsentanten des Systemverhaltens für das gesamte Betriebskontinuum sein. Häufig wird eine untere und obere Grenze für jeden unsicheren Parameter festgelegt. Die Eckpunkte des so entstehenden Hyperrechteckes (Parameterbox) sind dann die Betriebspunkte. Somit ergeben sich bei  $m$  unsicheren Parametern  $l=2^m$  Betriebspunkte. Zur Bestimmung des Gütekriteriums werden  $l$  Simulationen mit konstant gehaltenen Reglerparametern  $x$  und den  $l$  definierten unsicheren Parametersätzen durchgeführt. Abbildung 8.10 zeigt die so gebildeten Fehlerflächen unter Verwendung des in Abschnitt 8.4.2 beschriebenen Schrankenriteriums (siehe Abbildung 8.9). Aus diesen  $l$  ermittelten Gütekriterien wird durch Summierung ein Ersatzkriterium gebildet:

$$Q_{ges} = \sum_{i=1}^l Q_i. \quad (8.10)$$

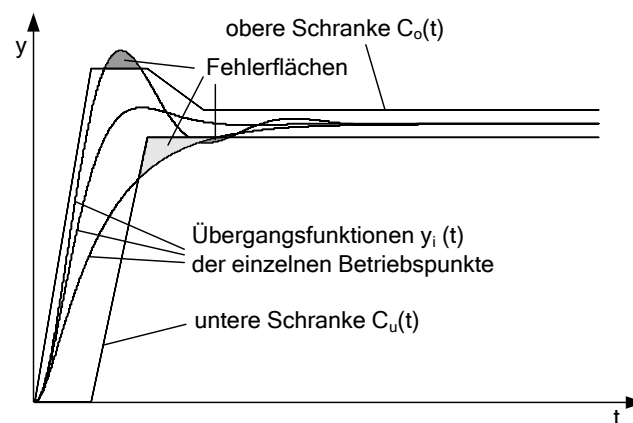


Abbildung 8.10: Bestimmung der einzelnen Gütekriterien



## 8.5 Robustheitsentwurf der Kursregelung

Nachfolgend wird der Robustheitsentwurf der Kursregelung unter MATLAB®/Simulink® vorgestellt. Die Auslegung des Reglers erfolgt hinsichtlich Führung und Störung für das AUV M63. Die unsicheren Parameter bei diesem Entwurf sind die Fahrgeschwindigkeit durchs Wasser und die Seeströmung. Tabelle 8.1 enthält die  $l$  definierten Betriebspunkte. Eine Variation der Strömungsrichtung ist nicht notwendig, da das Fahrzeug durch die vorgegebene Sollwertfolge des Kurses unterschiedliche Ausrichtungen zur Strömung besitzt.

Tabelle 8.1: Definition der Betriebspunkte

Betriebspunkt	Fahrgeschwindigkeit durchs Wasser in m/s	Strömungsvektor in m/s
Betriebspunkt 1	1	[0 0 0]
Betriebspunkt 2	2	[0 0 0]
Betriebspunkt 3	2,7	[0 0 0]
Betriebspunkt 4	1	[1,5 0 0]
Betriebspunkt 5	2	[1,5 0 0]
Betriebspunkt 6	2,7	[1,5 0 0]

Der zu optimierende adaptive PI-Regler wurde in Abschnitt 8.3.4 vorgestellt. Eine Anpassung der Reglerparameter  $KP$  und  $KI$  erfolgt in Abhängigkeit von der aktuellen Fahrgeschwindigkeit durchs Wasser. Zur Beschreibung dieses funktionellen Zusammenhangs wurden Geradengleichungen mit einer unteren und oberen Sättigung definiert, wodurch lediglich zwei Stützstellen zur Beschreibung dieser Funktion notwendig sind. Abbildung 8.11 zeigt eine solche Kennlinie für  $KP$ . Die Abszissenwerte der beiden Stützstellen enthalten die minimale und maximale Fahrgeschwindigkeit. Sie werden als konstant vorausgesetzt und brauchen nicht optimiert zu werden. Lediglich die dazugehörigen Ordinatenwerte  $KP_{links}$  und  $KP_{rechts}$  werden optimiert. So ergibt sich der zu optimierende Reglerparametervektor zu  $\mathbf{x} = [KP_{links}, KP_{rechts}, KI_{links}, KI_{rechts}]$ . Es ist natürlich auch möglich, Lookup-Tabellen oder ein Fuzzy-System zur Beschreibung des funktionellen Zusammenhangs einzusetzen, wodurch eine bessere Anpassung der Reglerparameter an die einzelnen Gebiete des Arbeitsbereiches erfolgt. Der Einsatz der Geradengleichung zeigte sich jedoch für den interessierenden Arbeitsbereich als ausreichend. Um geeignete Startparameter für die Optimierung zu ermitteln, wurden bei minimaler und maximaler Fahrgeschwindigkeit die Sprungantworten des Gierwinkelregelkreises als Übertragungsfunktionen approximiert und ihre Parameter ermittelt. Aus ihnen erfolgte unter Verwendung von Bemessungsregeln [128] die Berechnung der Reglerparameter für den Start.

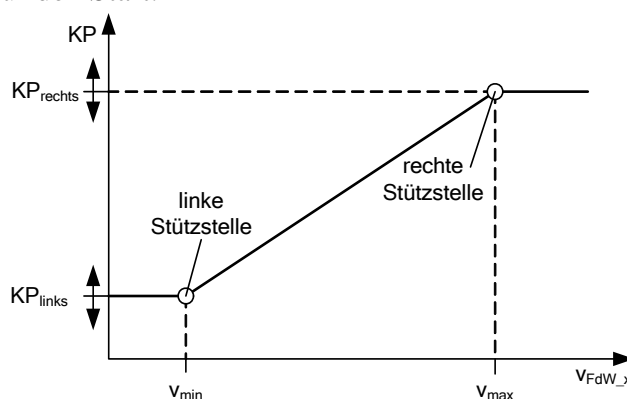


Abbildung 8.11: Geschwindigkeitsabhängige Kennlinie von  $KP$

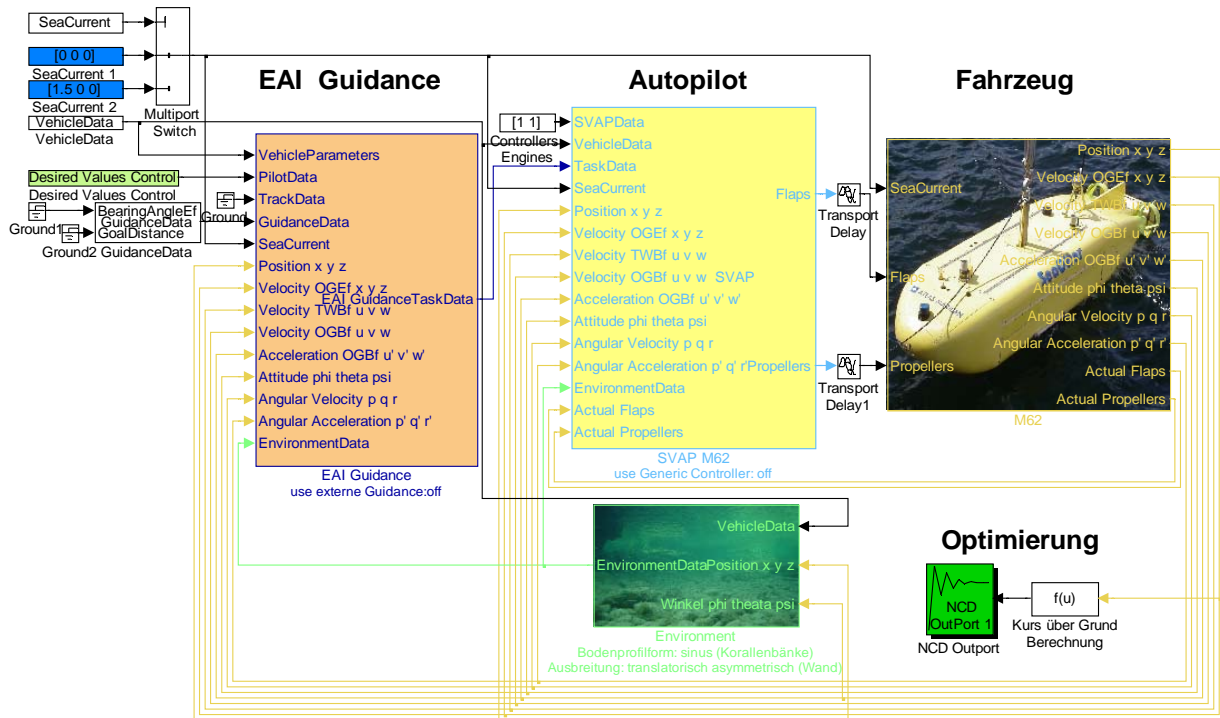


Abbildung 8.12: Simulink-Modell zur Optimierung der Kursregelung

In Abbildung 8.12 ist das Simulink-Modell zur Optimierung der Kursregelung dargestellt. Am Eingang des Blockes der Führungsalgorithmen liegt die definierte Sollwertfolge des Kurses an. Am Ausgang wird der Gierwinkelsollwert an den Autopiloten übergeben, der seinerseits die Aktuatorwerte für das Fahrzeugmodell generiert. Aus dem erdfesten Geschwindigkeitsvektor wird der Istkurs ermittelt und an den NCD-Block übergeben. Dieser Block berechnet aus den Kurssignalverläufen der einzelnen sechs Simulationen das Gütekriterium für das evolutionäre Optimierungsverfahren. Das Verhalten des optimierten Kursreglers ist aus den Tests der *Reaktiven Steuerung* und des *Identifikationssystems* zu ersehen.

## 8.6 Zusammenfassung

Die Aufgabe der Führungsalgorithmen besteht in der Generierung von Sollwerten für den Autopiloten zur Bewegungssteuerung des Fahrzeuges aus den Vorgaben der Submodule *Kollisionsvermeidungssystem* oder *Identifikationssystem*. Die Vorgaben lassen sich in vier Führungsaufgaben einteilen. Diese sind Kurs- und Bahnführung sowie Abstands- und Peilungssteuerung. Für jede dieser Aufgaben wurde ein Regler auf Basis eines adaptiven PID-Reglers entwickelt und simulativ getestet. Die Verwendung der Parameteroptimierung zur Ermittlung der Reglerparameter ermöglichte es auf einfache Weise, das nichtlineare Systemverhalten der unterlagerten Regelkreise sowie die Robustheitsanforderung an die Regelungen zu berücksichtigen. Als Parametersuchverfahren wurden Evolutionsstrategien eingesetzt, die eine hohe Robustheit gegenüber Unstetigkeiten im Gütegebirge besitzen und fähig sind, das globale Minimum zu lokalisieren. Die Gütewertberechnungen erfolgten durch Simulation des modellierten Regelkreises unter Einbeziehung des Autopiloten- und Fahrzeugverhaltens. Das verwendete Schrankenkriterium ermöglicht eine einfache Umsetzung der qualitativen Regelkreisanforderungen in einem Gütekriterium.

## 9 Test und Erprobung

In den vorhergehenden Kapiteln 2 bis 8 wurden die einzelnen Komponenten des Moduls *Fahrzeugführung in Sondersituationen* im Detail vorgestellt. Dabei erfolgten in einigen Kapiteln Laufzeituntersuchungen und Überprüfungen ihrer Funktionalität. Das Zusammenwirken der einzelnen Komponenten soll in diesem Kapitel anhand simulativer und praktischer Untersuchungen/Erprobungen dargelegt werden. Ausgehend von einer kurzen Vorstellung der entwickelten Simulations- und Testumgebung unter MATLAB®/Simulink® [142], wird mit ihr im ersten Teil dieses Kapitels das Verhalten des Moduls *FIS* in Ausweichsituationen untersucht. Bei diesen simulativen Untersuchungen erfolgt ein Vergleich der *Reaktiven Steuerung* und der *Wegeplanung* hinsichtlich der generierten Fahrtrajektorien bei einem vorgegebenen Hindernisparcours unter Einbeziehung der Seeströmung. Die Auswertung einer durchgeführten Mission, die in Herbst 2004 im Rahmen von Seeversuchen mit den AUV M62 in der Ostsee stattfand, wird im zweiten Teil behandelt.

### 9.1 Simulations- und Testumgebung unter MATLAB®/Simulink®

Für die Entwicklung und zum Test der einzelnen Algorithmen des Moduls *FIS* wurde unter MATLAB®/Simulink® eine Simulationsumgebung realisiert. Diese Umgebung ermöglicht:

- eine effiziente Fehlersuche durch die Einbindung der mit dem Borland C++ Builder geschriebenen Algorithmen des Moduls *FIS* in S-Funktionen unter Simulink®, (So werden die S-Funktionen im Borland C++ Builder als DLL (Dynamic-Link Library) generiert. Bei einer Fehlersuche kann dann vom Debug-Modus aus MATLAB® gestartet und ein Simulink-System, welches den Block des Moduls *FIS* enthält, abgearbeitet werden.)
- eine einfache Auswertung der Tests und Darstellung der Ergebnisse unter MATLAB®,
- die Optimierung der einzelnen Reglerparameter der *Führungsalgorithmen* nach nutzerspezifischen Anforderungen (siehe Abschnitt 8.4),
- das einfache und schnelle Testen des Moduls *FIS* bzw. seiner Submodule ohne Verwendung zusätzlicher Softwaremodule (Missionsüberwachung, Fahrzeugsimulator, Objekterkennung, HMI, CORBA, ...) des AUVs.

Eine Forderung an die Simulationsumgebung bestand in der unveränderten Einbindung des in C++ geschriebenen Codes des Moduls *FIS* in die Simulink®-eigenen Programme (S-Funktionen). Durch das verwendete Interface-Design des Moduls *FIS* war dies leicht möglich. Dabei existieren abstrakte Basisklassen, welche die öffentlichen Schnittstellen definieren. In diesen Klassen gibt es Referenzen mit den zu kommunizierenden Objekten. Zur Erzeugung der DLL für den *FIS*-Block wurde lediglich die Klasse der Datenschnittstelle von

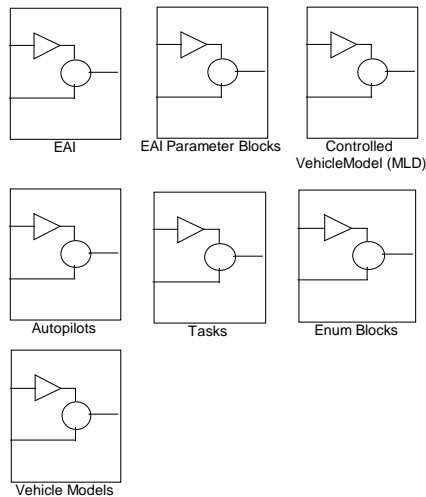


Abbildung 9.1: Blockbibliothek

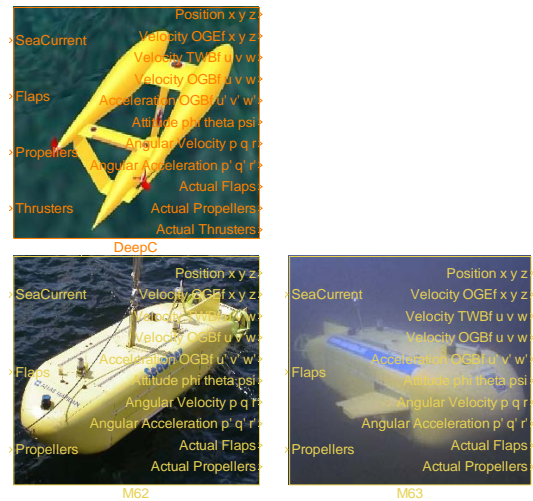


Abbildung 9.2: Fahrzeugmodelle

ihrer abstrakten Klasse abgeleitet, eine Klasse zur Umwandlung der Simulink®-Datenstrukturen in die *FIS*-Datenstrukturen geschrieben und die Objekte der einzelnen Submodule in die S-Funktion des *FIS*-Blockes implementiert. Beim Übersetzen wurden die einzelnen Bibliotheken des Moduls *FIS* dann mit eingebunden. So ist es möglich, Änderungen in der Funktionalität bzw. innerhalb des *FIS*-Codes durch ein erneutes Übersetzen des Simulink®-Programms einfach zu übernehmen. Die entwickelte Simulationsumgebung enthält eine Blockbibliothek (siehe Abbildung 9.1), die neben dem *FIS*-Block die Simulationsblöcke für den *Autopiloten*, die Fahrzeugmodelle (siehe Abbildung 9.2), ein Umweltmodell sowie Parameter- und Aufgabenblöcke umfasst. (Bemerkung: In der Blockbibliothek steht die englische Projektbezeichnung für *FIS*. Diese lautet EAI, was *Evasion And Identification* bedeutet.)

## 9.2 Simulative Untersuchung der Kollisionsvermeidung

Zur Untersuchung der Kollisionsvermeidung des Moduls *FIS* wurde das in Abbildung 9.3 dargestellte Simulink®-Modell für das AUV M62 gebildet. Durch die mit diesem System durchgeführten Simulationen konnten die einzelnen Submodule von der *Zielpunktgenerierung* über das *Kollisionsvermeidungssystem* bis zu den *Führungsalgorithmen* im Zusammenwirken getestet werden. Bei den durchgeführten Tests wurde der in Abbildung 9.4 dargestellte Hindernisparcours verwendet, wobei von unterschiedlichen Positionen aus gestartet wurde. Bei den Untersuchungen erfolgte ein Vergleich der *Wegeplanung* und der *Reaktiven Steuerung*. Tabelle 9.1 stellt die Ergebnisse dar.

Tabelle 9.1: Testszenarien

	Art der Kollisionsvermeidung	Weglänge in m		Zeit zum Abfahren des Weges in s	
		$v_{\text{Strömung}} = [0 \ 0 \ 0] \text{ m/s}$	$v_{\text{Strömung}} = [0,7 \ 0,7 \ 0] \text{ m/s}$	$v_{\text{Strömung}} = [0 \ 0 \ 0] \text{ m/s}$	$v_{\text{Strömung}} = [0,7 \ 0,7 \ 0] \text{ m/s}$
Startpunkt 1	Reaktive Steuerung	242,04 m	240,39 m	120,3 s	88,6 s
	Wegeplanung	233,43 m	233,63 m	116,7 s	84 s
Startpunkt 2	Reaktive Steuerung	239,47 m	255,29 m	118,9 s	109,5 s
	Wegeplanung	231,34 m	233,1 m	115,6 s	92,4 s
Startpunkt 3	Reaktive Steuerung	250,05 m	246,56 m	124,3 s	108,3 s
	Wegeplanung	242,06 m	244,83 m	121 s	110,7 s



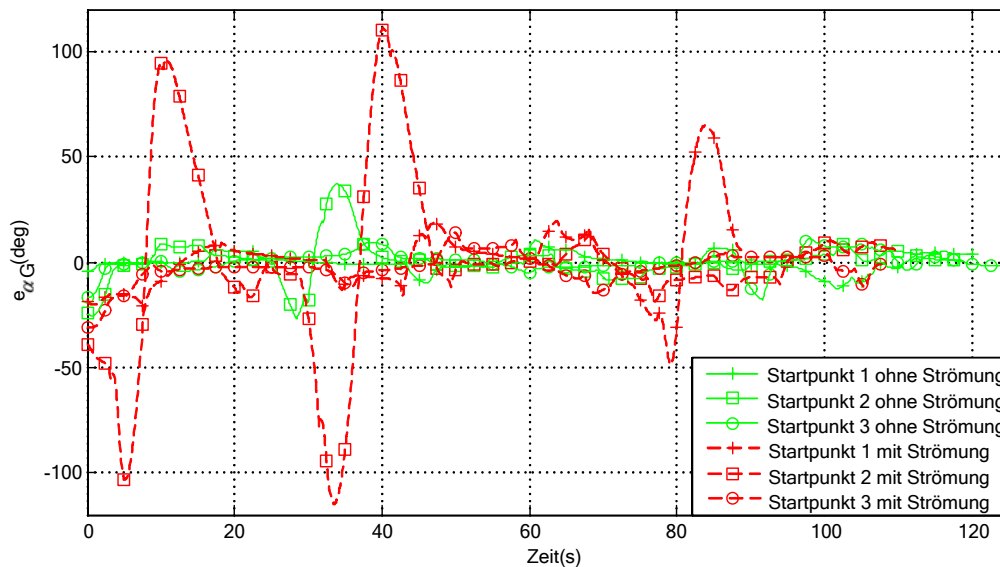
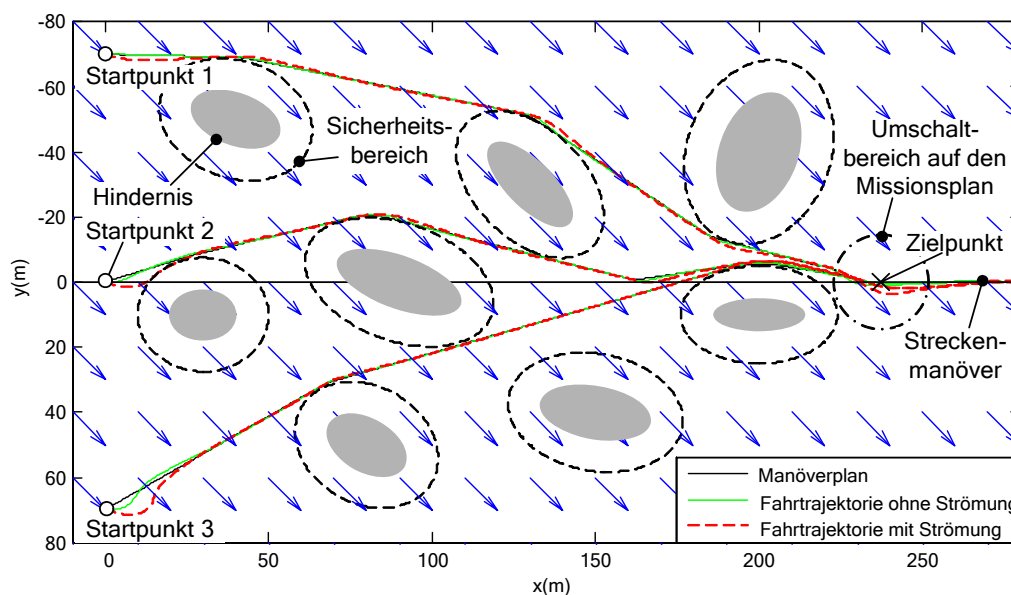
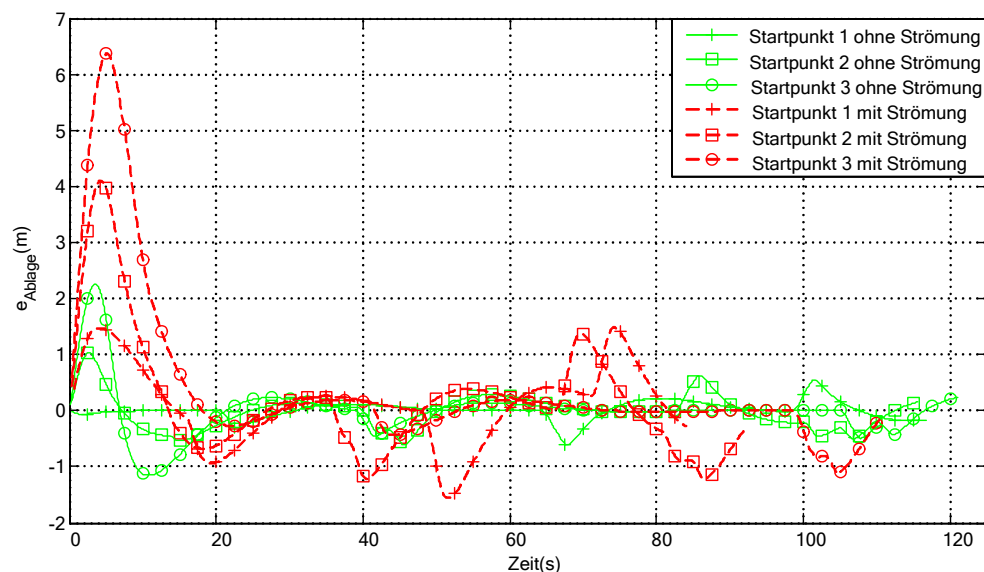
Abbildung 9.5: Regelabweichungen  $e_{\alpha G}$  der Kursregelung

Abbildung 9.5 zeigt die Regelabweichungen für die einzelnen Simulationen mit und ohne Strömung. Dabei sind drei extreme Abweichungen zu beobachten. Die erste Abweichung entsteht durch die oben beschriebene Initialisierung des Kursreglers bei Startpunkt 2 und einer vorhandenen Seeströmung. Das zweite extreme Kursmanöver bei dieser Simulation erklärt sich aus dem reaktiven Verhalten des Kollisionsvermeidungsalgorithmus in Kombination mit der vorhandenen Seeströmung. Da der generierte Sollkurs nur unter Verwendung des zum Fahrzeug nächstgelegenen Hindernisses bestimmt wird, umfährt das Fahrzeug das erste Hindernis und nimmt Kurs auf den Zielpunkt mit einer Geschwindigkeit über Grund von ca. 2,7 m/s. Beim Einfahren in den Übergangsbereich zum zweiten Hindernis (siehe Abschnitt 5.3.4) ändert sich der Sollkurs durch die hohe Fahrgeschwindigkeit und die Größe des Hindernisses sehr rasch. Dies ist auch die Ursache der großen Regelabweichung bei Startpunkt 1. Vergleicht man die benötigten Weglängen und die Zeiten zum Abfahren des Hindernisparcours in Tabelle 9.1, so zeigt die *Reaktive Steuerung* ähnliche Ergebnisse wie die *Wegeplanung*, was für einen reaktiven Ansatz bemerkenswert ist.

Abbildung 9.6: Fahrtrajektorien durch einen Hindernisparcours bei Verwendung der *Wegeplanung*



Abbildung 9.7: Bahnablagen  $e_{\text{Ablage}}$  der Bahnregelung

Bei der zweiten Testreihe wurde der Hindernisparcours unter Verwendung des hybriden *Kollisionsvermeidungssystems* durchfahren. Abbildung 9.6 zeigt die einzelnen Fahrtrajektorien bei den verschiedenen Startpositionen. Da die Generierung des Manöverplanes schon nach ca. 200 ms abgeschlossen ist, ist das Aktivsein der *Reaktiven Steuerung* während dieser ersten 200 ms in den Trajektorienverläufen kaum zu erkennen, so dass die Verläufe nur durch die *Wegeplanung* bestimmt werden. Bei einer vorhandenen Seeströmung muss der Bahnregler nach seiner Initialisierung die Störeinflüsse der Seeströmung durch seinen Integralanteil kompensieren. So entstehen die in Abbildung 9.7 dargestellten großen Ablagen zu Beginn der Simulationen. Danach fährt das Fahrzeug, wie „auf Schienen geführt“, die vorgegebenen Manöverpläne in den einzelnen Simulationen ab. Die maximalen Ablagen betragen bei den strömungsfreien Simulationen ca. 0,5 m, die beim Einfahren in die Kreisbogenmanöver entstehen. Dies liegt daran, dass die Soll-drehrate beim Abfahren eines Streckenmanövers null ist und beim Einfahren in ein Kreisbogenmanöver plötzlich einen Wert annimmt. Diesem Wert kann das Fahrzeug nur verzögert folgen, was sich in den entstehenden Ablagen widerspiegelt. Hier bräute die Verwendung von interpolierten Trajektorien (Splinekurven) einen Vorteil, da es dort keine diskontinuierlichen Änderungen der Sollkursraten gibt [58]. Das heißt, die zweiten Ableitungen der Splinekurven besitzen keine Unstetigkeiten. Die generierten Manöverpläne zeigen einen optimalen Verlauf, der dadurch charakterisiert ist, dass die einzelnen Streckenmanöver als Tangenten um die Sicherheitsbereiche herumgeführt werden. Der durch die *Zielpunktgenerierung* berechnete Zielpunkt liegt weit genug von den Objekten entfernt, so dass das Fahrzeug kollisionsfrei auf das Streckenmanöver des Missionsplanes einfahren kann.

## 9.3 Seeveruche

Im Herbst 2004 fanden in der Eckernförder Bucht Seeveruche mit dem AUV M62 statt. Ziel der Versuche war die Überprüfung der entwickelten Algorithmen zur aktiven Fahrzeugautonomie. In diesem Zusammenhang wurden auch die Algorithmen des Moduls *FIS* erprobt. Dazu wurden in einem abgesperrten Gebiet künstliche Hindernisse ausgebracht und ein Missionsplan durch die einzelnen Hindernispositionen gelegt. Abbildung 9.8 zeigt das Aussetzen des AUVs M62 zur Durchführung der Mission. Der Missionsplan sowie die geloggte Fahrtrajektorie sind in der operationellen Bedienoberfläche in Abbildung 9.9 dargestellt.



Abbildung 9.8: Aussetzen des AUVs M62 [9]

Abbildung 9.10 zeigt die Fahrtrajektorie und die Hindernispositionen. Alle vorhandenen Hindernisse wurden von der *Objekterkennung* erfasst und konnten durch das Modul *FIS* umfahren werden. Das relativ späte Ausweichen erklärt sich durch die gering gewählte Sonarreichweite und die Zeit zum Verifizieren der detektierten Objekte in der *Objekterkennung*.

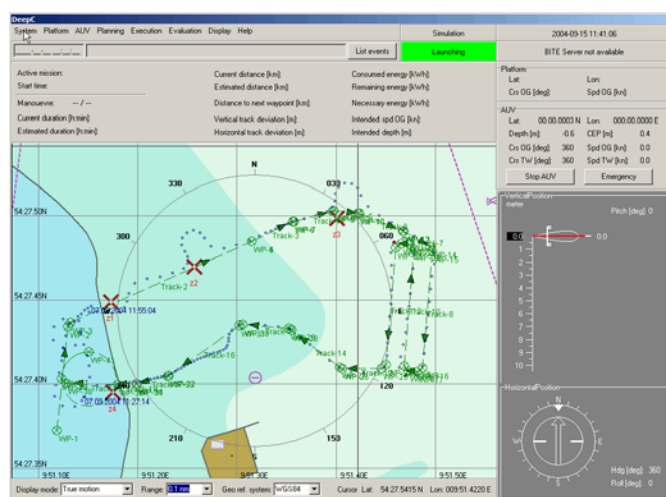


Abbildung 9.9: Erstellter Missionsplan mit der geloggtten Fahrtrajektorie [9]

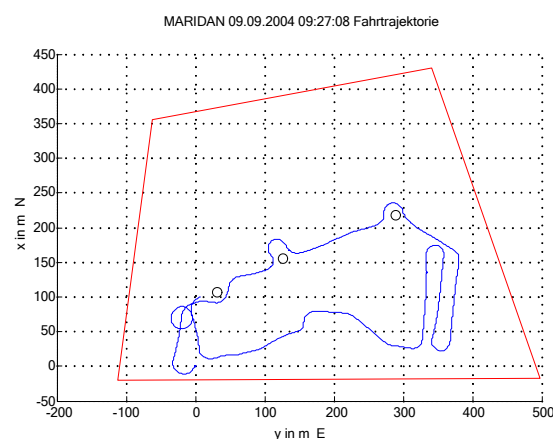


Abbildung 9.10: Fahrtrajektorie mit den korrekten Hindernispositionen

## 9.4 Zusammenfassung

In diesem Kapitel wurde die Simulations- und Testumgebung unter MATLAB<sup>®</sup>/Simulink<sup>®</sup> im Überblick vorgestellt sowie ihre Einsatzmöglichkeiten bei der Simulation aufgezeigt. Diese Umgebung war während der Projektlaufzeit ein effizientes Hilfsmittel bei der Erstellung, beim Test und bei der Fehlersuche der Algorithmen und des Codes des Moduls *FIS*. Die in diesem Kapitel vorgestellten Tests wurden mit dem AUV M62 durchgeführt. So konnte die entwickelte Software des Moduls *FIS* auch mit anderen AUVs getestet werden. Die durchgeführten Simulationen zur Kollisionsvermeidung zeigen die Leistungsfähigkeit der entwickelten Algorithmen, die auch bei einer vorhandenen Seeströmung sicher funktionieren. Die erfolgreiche praktische Erprobung der Algorithmen bei Seeversuchen schließt den Kreis von den ersten Literaturstudien über das Entwickeln der Algorithmen, die simulativen Untersuchungen bis hin zu einer getesteten Software zur Führung eines Autonomen Unterwasserfahrzeuges in Sondersituationen.



## 10 Zusammenfassung und Ausblick

Die vorliegende Arbeit wurde auf Basis der Anforderungen, Vorgaben und Bedingungen zur Führung autonomer Unterwasserfahrzeuge, vom Entwurf bis zu den abgeschlossenen Tests der Algorithmen bei Seeversuchen, entwickelt. Ein Großteil der Arbeit bestand im objektorientierten Entwurf, der Implementierung und den Tests der Algorithmen im Borland C++ Builder. Der entwickelte Code enthält neben den eigentlichen Algorithmen Funktionen zur Fehlerüberwachung sowie zum Daten- und Nachrichtenloggen. Die Beschreibung der softwaretechnischen Details blieb in der vorliegenden Arbeit weitestgehend unberücksichtigt, da dies den Rahmen sprengen würde. Die Arbeit enthält die Beschreibung der neu entwickelten Algorithmen und die erzielten Ergebnisse.

### 10.1 Zusammenfassung

Die Arbeit stellt ein vollständiges neues Konzept zur Fahrzeugführung in Sondersituationen für ein Autonomes Unterwasserfahrzeug vor. Eine Sondersituation ist dann gegeben, wenn während einer Mission Objekte den abzufahrenden Routenplan tangieren. Die möglichen Handlungen bestehen im Ausweichen oder in der Identifikation dieser Objekte. Bei der Erstellung des Konzeptes gab es eine Reihe praxisrelevanter Anforderungen (Sicherheit, Robustheit, Rechenzeit, Optimalität) an die zu entwickelnden Verfahren und Algorithmen, die unter den hard- und softwaretechnischen Vorgaben und Arbeitsbedingungen einzuhalten waren. Solche Vorgaben umfassen das nichtholonome, verzögerte Bewegungsverhalten des Unterwasserfahrzeuges, die Sensorik zur Bestimmung der Umwelt, die im Fahrzeug eingesetzte Rechentechnik sowie die zu verwendende Entwicklungs- und Kommunikationssoftware. Die Arbeitsbedingungen eines Unterwasserfahrzeuges sind durch ein Manövrieren im dreidimensionalen Raum bei einer möglichen Seeströmung, schlechter Sicht und hohem Wasserdruck unter Einhaltung eines Sicherheitsabstandes zum Meeresboden und zu den Hindernissen charakterisiert. Solche Hindernisse können geographische Hindernisse (Wände, Erhebungen), technische Bauten (Trägerkonstruktionen, Kaimauern) oder Altlasten (Container, Minen, Schiffwracks) sein. Das entwickelte Konzept ist modular aufgebaut und umfasst Komponenten/Submodule zur Kollisionsüberwachung, Zielpunktgenerierung, Kollisionsvermeidung, Fahrzeugführung bei Identifikationsaufgaben sowie zur Fahrzeugsteuerung.

Für die Kollisionserkennung wird ein mehrstufiges Verfahren zur effizienten Bestimmung der Kollisionssituation zwischen den elliptischen Objektzylindern und den Strecken- und Kreisbogenmanövern vorgestellt. Das Verfahren basiert auf einem Zwei-Phasen-Konzept. Dieses Konzept ermöglicht eine schnelle Kollisionsüberprüfung durch die Verwendung einfacher Kollisionstests zur Vorselektion möglicher Kollisionskandidaten. Es wird auch eingesetzt für die:

- Überprüfung möglicher Verbindungen zwischen den Knoten bei der Erstellung des Sichtbarkeitsgraphen,
- Glättung des Routenplanes,
- Überprüfung des Manöverplanes und
- Bestimmung des Zielpunktes.

Das in dieser Arbeit entwickelte Kollisionsvermeidungssystem besitzt eine hybride Struktur, bei der ein reaktiver (Reaktive Steuerung) und ein planender Ansatz (Wegeplanung) parallel arbeiten. Die Reaktive Steuerung übernimmt die Führung des Fahrzeuges, während die Wegeplanung einen Routenplan generiert. Steht ein Routenplan zur Verfügung, arbeitet die Wegeplanung diesen ab.

Für die Reaktive Steuerung wurde ein neues Verfahren zur geometrischen Konstruktion von Gradientenlinien entwickelt. Es verbindet die Vorteile des von Guldner entwickelten Verfahrens der Harmonischen Dipolpotentiale mit der Forderung einer wegoptimalen Fahrweise unter Beachtung des nichtholonomen, verzögerten Bewegungsverhaltens des Fahrzeuges.

Durch die Forderung der Online-Erzeugung eines Manöverplanes unter Gewährleistung des Echtzeitverhaltens des Systems wurden graphenbasierte Verfahren für die Wegeplanung in dieser Arbeit untersucht. Diese Verfahren ermöglichen es, einen optimalen Weg nach definierten Vorgaben in einer kalkulierbaren Zeit zu ermitteln. Für die Erstellung eines geometrischen Graphen aus einem Konfigurationsraum mit elliptischen Objektzylindern wurden die entwickelten Verfahren des Quadtree- / Octreegraphen und des sektorisierten Sichtbarkeitsgraphen vorgestellt. Die Einbeziehung der Seeströmung in die Kostenfunktion ermöglicht es, Gebiete mit starker Seeströmung von der Routenplanung auszuschließen. Bei der Routensuche kamen der Dijkstra- und der A\*-Algorithmus zum Einsatz. Der Einsatz des A\*-Algorithmus führt zu einer Senkung der Berechnungszeit gegenüber dem Dijkstra-Algorithmus, ohne dabei zu anderen Ergebnissen bei der Routensuche zu führen. Eine signifikante Senkung der Gesamtrechnenzeit bei der Erzeugung eines Manöverplanes kann jedoch nur mit einer Verringerung der Rechenzeit zur Generierung des geometrischen Graphen einhergehen. Da die in dieser Arbeit entwickelten Verfahren zur Graphenerzeugung keine Graphen generieren, bei welchen alle möglichen Verbindungen zwischen den Knoten durch Kanten beschrieben sind, zeigen die gefundenen Routen einen stufen- bzw. schlangenförmigen Verlauf. Deshalb wurde ein neues Verfahren entwickelt, welches diese Routenpläne unter Einhaltung der Kollisionsfreiheit mit den Objekten glättet. Damit wird ein Routenplan erzeugt, der das dynamische Verhalten der Fahrzeuge berücksichtigt.

Bei der Durchführung einer Mission kann die Identifikation von Objekten eine geforderte Aufgabe sein. In Abhängigkeit von den vorgegebenen Objektmaßen werden die Arbeitsmodi *Ausrichten auf ein Objekt* und *Umfahren eines Objektes* unterschieden. Die einzelnen Modi setzen sich aus mehreren Manövern bzw. Führungsaufgaben zusammen, die abhängig von der Seeströmung und der Fahrzeugposition zum identifizierenden Objekt aktiviert werden.

Um das Fahrzeug nach einer Aktivierung des Moduls *FIS* zur Erfüllung einer Sondersituation wieder auf den Missionsplan zurückzuführen, ist die Bestimmung eines anzufahrenden Zielpunktes notwendig. Dieser Zielpunkt wird in der Zielpunktgenerierung so berechnet, dass ein kollisionsfreies Einfahren auf die aktuelle Sollbahn des Missionsplanes gewährleistet ist.

Die einzelnen Führungsaufgaben wurden in separaten Reglern implementiert, die als adaptive PID-Regler aufgebaut sind. Die Verwendung der Parameteroptimierung zur Ermittlung ihrer Reglerparameter ermöglichte es auf einfache Weise, das nichtlineare Systemverhalten des Fahrzeuges sowie die Robustheitsanforderungen an die Regelungen zu berücksichtigen.

Die vorgestellte Simulations- und Testumgebung unter MATLAB®/Simulink® war ein effizientes Hilfsmittel bei der Erstellung, beim Test und bei der Fehlersuche der Algorithmen und des Codes des Moduls *FIS*.

## 10.2 Ausblick

Die methodisch und praxisorientierte Aufgabenstellung dieser Arbeit und die zeitlichen Restriktionen (dreijährige Laufzeit des BMBF-Projektes DeepC) führten zu einer großen Motivation bei der Entwicklung und Implementierung der Algorithmen. Andererseits konnten viele Ideen, mögliche neue Ansätze sowie die Erweiterung der Algorithmen zur Erfüllung neuer Anforderungen, welche sich aus Veröffentlichungen und eigenen Überlegungen ergaben, noch nicht behandelt werden. Es sei jedoch erwähnt, dass alle zu Projektbeginn definierten Anforderungen mit dieser Arbeit voll erfüllt wurden.

Ein möglicher Untersuchungsschwerpunkt für künftige Arbeiten ist die Untersuchung von *Sweep and Prune Algorithmen* zur Kollisionsüberwachung, wenn größere Objekt- bzw. Manöverbereiche überprüft werden müssen.

Zur Erstellung eines geometrischen Graphen auf Basis von elliptischen Objektzylindern wird in kommenden Arbeiten der Kreistangentenansatz untersucht [115]. Ziel der Untersuchungen ist die Senkung der Rechenzeit zur Generierung des Graphen. Die Online-Generierung eines Routenplanes unter Einbeziehung von bewegten Objekten stellt einen weiteren Forschungsschwerpunkt dar.

Die Einbeziehung der Strömungsinformation bei der Bestimmung des Kostenwertes für die Kanten des Graphen wurde in dieser Arbeit vorgestellt. Dabei geht man davon aus, dass der am Fahrzeug ermittelte Strömungswert im gesamten Operationsgebiet gleich ist. Da die räumlichen Strömungsverläufe um Hindernisse (vor und hinter dem Hindernis ist die Strömung gering (Strömungsschatten), an den Seiten dagegen ist sie groß), in der Nähe des Meeresbodens (gefährliche Grundströmungen) und in der Nähe von Wänden unterschiedlich sein können, sollten diese Informationen in ein Modell zur Bestimmung eines räumlichen Strömungsfeldes einbezogen werden. So kann die Wegeplanung diese Informationen berücksichtigen, ähnlich einem Taucher, der die örtlichen Gegebenheiten in strömungsreichen Gebieten nutzt, um kräftesparend und gefahrenfrei tauchen zu können [132].

Für die Führungsalgorithmen sollten ferner neue Regelungsmethoden, wie sie in [58] vorgestellt werden, erprobt und miteinander verglichen werden.

# Anhang

## A.1 Mathematische Beziehungen

### A.1.1 Quadratische Gleichung der Ellipse

Nachfolgend wird die Herleitung der quadratischen Gleichung der Ellipse vorgestellt. Eine achsenparallele Ellipse kann durch ihre Mittelpunktgleichung in der Form:

$$Q(x, y) = \frac{x^2}{a^2} + \frac{y^2}{b^2} - 1 = 0 \quad (\text{A.1})$$

bzw. 
$$b^2 x^2 + a^2 y^2 - a^2 b^2 = 0 \quad (\text{A.2})$$

beschrieben werden. Die Parameter  $a$  und  $b$  bezeichnen die Haupt- und Nebenachse der Ellipse. Bei einer Drehung der Ellipse um den Winkel  $\theta$  werden die  $x$  und  $y$  Koordinaten unter Verwendung der Rotationsmatrix  $\mathbf{R}_\theta$  transformiert:

$$\begin{bmatrix} x_{rot} \\ y_{rot} \end{bmatrix} = \underbrace{\begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}}_{\mathbf{R}_\theta} \begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \underbrace{\begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}}_{\mathbf{R}_\theta^T} \begin{bmatrix} x_{rot} \\ y_{rot} \end{bmatrix}. \quad (\text{A.3})$$

Durch Einsetzen der Transformationsbeziehungen von  $x$  und  $y$ :

$$\begin{aligned} x &= \cos \theta x_{rot} + \sin \theta y_{rot} \\ y &= -\sin \theta x_{rot} + \cos \theta y_{rot} \end{aligned} \quad (\text{A.4})$$

in die Gleichung (A.2) ergibt sich die quadratische Gleichung einer im Koordinatenursprung liegenden und gedrehten Ellipse zu:

$$\begin{aligned} &b^2 (\cos \theta x + \sin \theta y)^2 + a^2 (-\sin \theta x + \cos \theta y)^2 - a^2 b^2 = 0 \\ &b^2 \cos^2 \theta x^2 + 2b^2 \cos \theta \sin \theta xy + b^2 \sin^2 \theta y^2 + a^2 \sin^2 \theta x^2 - a^2 2 \sin \theta \cos \theta xy + a^2 \cos^2 \theta y^2 - a^2 b^2 = 0 \quad (\text{A.5}) \\ &x^2 \underbrace{(a^2 \sin^2 \theta + b^2 \cos^2 \theta)}_{a_{\text{Ellipse}}^2} + xy \underbrace{(2 \sin \theta \cos \theta (b^2 - a^2))}_{b_{\text{Ellipse}}} + y^2 \underbrace{(a^2 \cos^2 \theta + b^2 \sin^2 \theta)}_{c_{\text{Ellipse}}} + \underbrace{-a^2 b^2}_{f_{\text{Ellipse}}} = 0. \end{aligned}$$

Wird die Ellipse um eine Position  $\mathbf{x}_c$  vom Koordinatenursprung aus verschoben:

$$\begin{bmatrix} x_{trans} \\ y_{trans} \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} x_c \\ y_c \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_{trans} \\ y_{trans} \end{bmatrix} - \begin{bmatrix} x_c \\ y_c \end{bmatrix} \quad (\text{A.6})$$

so können durch Einsetzen der Transformationsbeziehungen von  $x$  und  $y$ :

$$\begin{aligned} x &= x_{trans} - x_c \\ y &= y_{trans} - y_c \end{aligned} \quad (\text{A.7})$$

in die Gleichung (A.5) die quadratische Gleichung der Ellipse zu:

$$\begin{aligned}
& (x - x_c)^2 (a^2 \sin^2 \theta + b^2 \cos^2 \theta) + \\
& (x - x_c)(y - y_c) (2 \sin \theta \cos \theta (b^2 - a^2)) + \\
& (y - y_c)^2 (a^2 \cos^2 \theta + b^2 \sin^2 \theta) + \\
& -a^2 b^2 = 0 \\
& (x^2 - 2xx_c + x_c^2) (a^2 \sin^2 \theta + b^2 \cos^2 \theta) + \\
& (xy - xy_c - x_c y + x_c y_c) (2 \sin \theta \cos \theta (b^2 - a^2)) + \\
& (y^2 - 2yy_c + y_c^2) (a^2 \cos^2 \theta + b^2 \sin^2 \theta) + \\
& -a^2 b^2 = 0 \\
& x^2 \underbrace{(a^2 \sin^2 \theta + b^2 \cos^2 \theta)}_{a_{\text{Ellipse}}} + \\
& xy \underbrace{(2 \sin \theta \cos \theta (b^2 - a^2))}_{b_{\text{Ellipse}}} + \\
& y^2 \underbrace{(a^2 \cos^2 \theta + b^2 \sin^2 \theta)}_{c_{\text{Ellipse}}} + \\
& x \underbrace{\left( -2x_c \underbrace{(a^2 \sin^2 \theta + b^2 \cos^2 \theta)}_{a_{\text{Ellipse}}} - y_c \underbrace{(2 \sin \theta \cos \theta (b^2 - a^2))}_{b_{\text{Ellipse}}} \right)}_{d_{\text{Ellipse}}} + \\
& y \underbrace{\left( -2y_c \underbrace{(a^2 \cos^2 \theta + b^2 \sin^2 \theta)}_{c_{\text{Ellipse}}} - x_c \underbrace{(2 \sin \theta \cos \theta (b^2 - a^2))}_{b_{\text{Ellipse}}} \right)}_{e_{\text{Ellipse}}} + \\
& \underbrace{x_c^2 \underbrace{(a^2 \sin^2 \theta + b^2 \cos^2 \theta)}_{a_{\text{Ellipse}}} + x_c y_c \underbrace{(2 \sin \theta \cos \theta (b^2 - a^2))}_{b_{\text{Ellipse}}} + y_c^2 \underbrace{(a^2 \cos^2 \theta + b^2 \sin^2 \theta)}_{c_{\text{Ellipse}}} - a^2 b^2}_{f_{\text{Ellipse}}} = 0
\end{aligned} \tag{A.8}$$

### A.1.2 Lösung eines Systems aus zwei Variablen und zwei quadratischen Gleichungen

Die nachfolgende Herleitung wurde aus [42] entnommen. Zwei quadratische Gleichungen:

$$\begin{aligned}
F(x, y) &= \alpha_{00} + \alpha_{10}x + \alpha_{01}y + \alpha_{20}x^2 + \alpha_{11}xy + \alpha_{02}y^2 = 0 \\
G(x, y) &= \beta_{00} + \beta_{10}x + \beta_{01}y + \beta_{20}x^2 + \beta_{11}xy + \beta_{02}y^2 = 0
\end{aligned} \tag{A.9}$$

können als Polynom für die Variable  $x$  durch die nachfolgenden Gleichungen:

$$\begin{aligned}
f(x) &= (\alpha_{20})x^2 + (\alpha_{11}y + \alpha_{10})x + (\alpha_{02}y^2 + \alpha_{01}y + \alpha_{00}) = a_2x^2 + a_1x + a_0 = 0 \\
g(x) &= (\beta_{20})x^2 + (\beta_{11}y + \beta_{10})x + (\beta_{02}y^2 + \beta_{01}y + \beta_{00}) = b_2x^2 + b_1x + b_0 = 0
\end{aligned} \tag{A.10}$$

beschrieben werden. Durch Aufstellen der Beziehungen:

$$a_2 f(x) - b_2 g(x) = (a_2 b_1 - a_1 b_2)x + (a_2 b_0 - a_0 b_2) = 0 \quad (\text{A.11})$$

$$b_1 f(x) - a_1 g(x) = (a_2 b_1 - a_1 b_2)x^2 + (a_0 b_1 - a_1 b_0) = 0 \quad (\text{A.12})$$

und Lösen der Gleichung (A.11) nach  $x$ :

$$x = \frac{(a_0 b_2 - a_2 b_0)}{(a_2 b_1 - a_1 b_2)} \quad (\text{A.13})$$

und Einsetzen in die Gleichung (A.12), ergibt sich die Beziehung:

$$\begin{aligned} (a_2 b_0 - a_0 b_2)^2 + (a_2 b_1 - a_1 b_2)(a_0 b_1 - a_1 b_0) &= 0 \\ (a_2 b_1 - a_1 b_2)(a_1 b_0 - a_0 b_1) - (a_2 b_0 - a_0 b_2)^2 &= 0 \end{aligned} \quad (\text{A.14})$$

Durch Auflösen dieser Gleichung und Einsetzen der Beziehungen für  $a_0, a_1, a_2$  und  $b_0, b_1, b_2$  von Gleichung (A.10) ergibt sich ein Polynom 4. Ordnung für die  $y$  Variable:

$$(a_2 b_1 - a_1 b_2)(a_1 b_0 - a_0 b_1) - (a_2 b_0 - a_0 b_2)^2 = \sum_{i=0}^4 h_i y^i = h(y) \quad (\text{A.15})$$

wobei

$$\begin{aligned} h_0 &= d_{00}d_{10} - d_{20}^2 \\ h_1 &= d_{01}d_{10} + d_{00}d_{11} - 2d_{20}d_{21} \\ h_2 &= d_{01}d_{11} + d_{00}d_{12} - d_{21}^2 - 2d_{20}d_{22} \\ h_3 &= d_{01}d_{12} + d_{00}d_{13} - 2d_{21}d_{22} \\ h_4 &= d_{01}d_{13} - d_{22}^2 \end{aligned} \quad (\text{A.16})$$

mit

$$\begin{aligned} d_{00} &= \alpha_{20}\beta_{10} - \beta_{20}\alpha_{10} \\ d_{01} &= \alpha_{20}\beta_{11} - \beta_{20}\alpha_{11} \\ d_{10} &= \alpha_{10}\beta_{00} - \beta_{10}\alpha_{00} \\ d_{11} &= \alpha_{11}\beta_{00} + \alpha_{10}\beta_{01} - \beta_{11}\alpha_{00} - \beta_{10}\alpha_{01} \\ d_{12} &= \alpha_{11}\beta_{01} + \alpha_{10}\beta_{02} - \beta_{11}\alpha_{01} - \beta_{10}\alpha_{02} \\ d_{13} &= \alpha_{11}\beta_{02} - \beta_{11}\alpha_{02} \\ d_{20} &= \alpha_{20}\beta_{00} - \beta_{20}\alpha_{00} \\ d_{21} &= \alpha_{20}\beta_{01} - \beta_{20}\alpha_{01} \\ d_{22} &= \alpha_{20}\beta_{02} - \beta_{20}\alpha_{02} \end{aligned} \quad (\text{A.17})$$

ist. Durch Bestimmung der Nullstellen der Gleichung (A.15) erhält man die gesuchten  $y$  Werte des Gleichungssystems. Diese werden zur Ermittlung der Parameter  $a_0, a_1, a_2$  und  $b_0, b_1, b_2$  von Gleichung (A.10) verwendet, welche in Gleichung (A.13) zur Berechnung der entsprechenden  $x$  Werte eingesetzt werden.



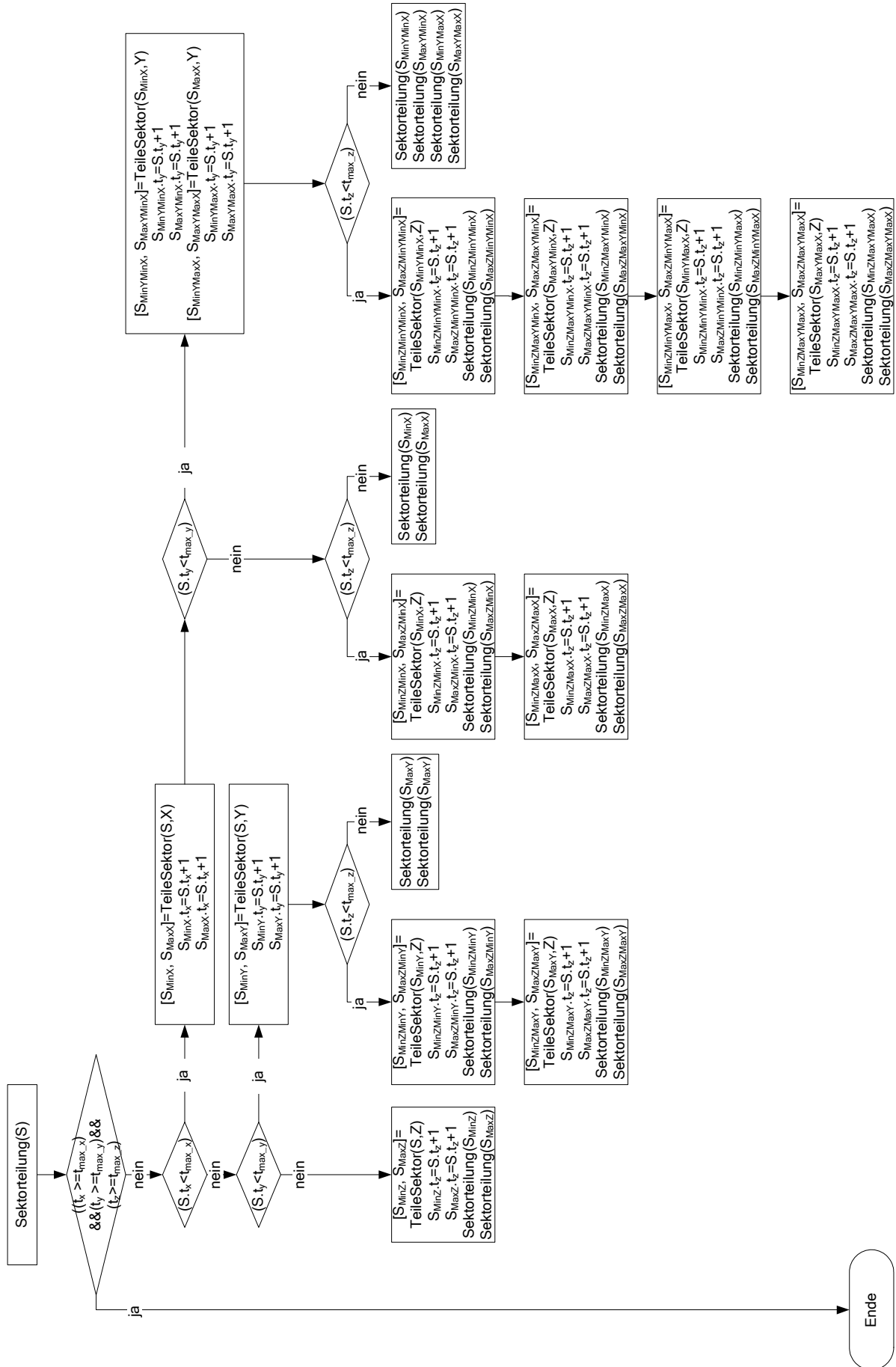
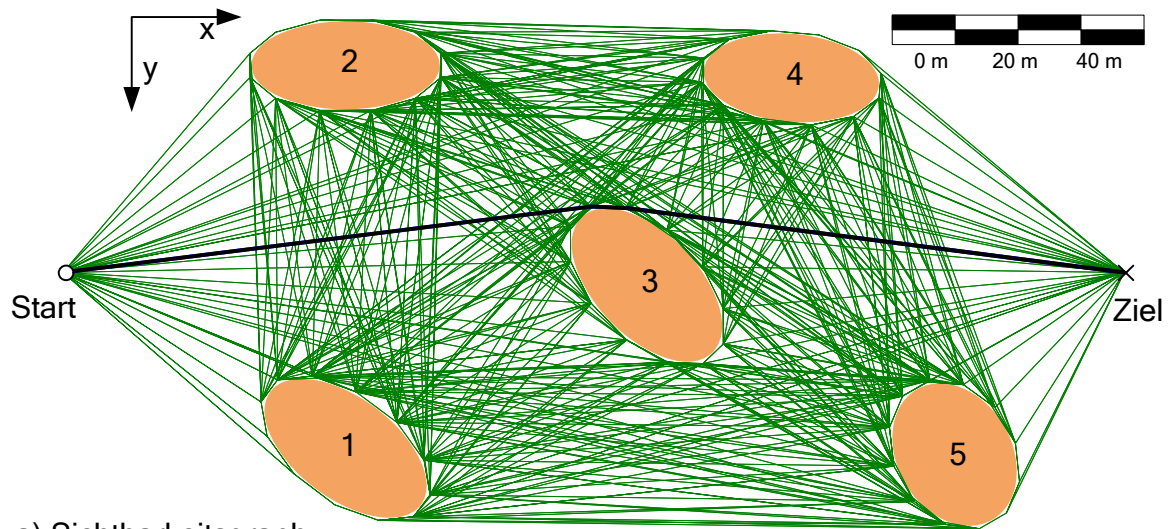


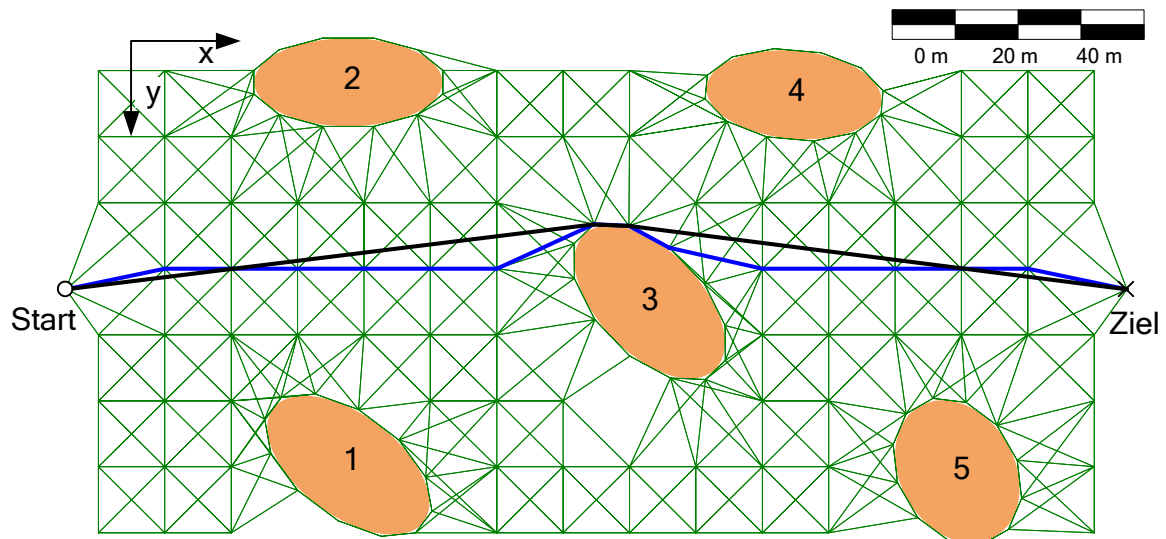
Abbildung A.2: Flussdiagramm zur Sektorzerlegung für den dreidimensionalen Raum



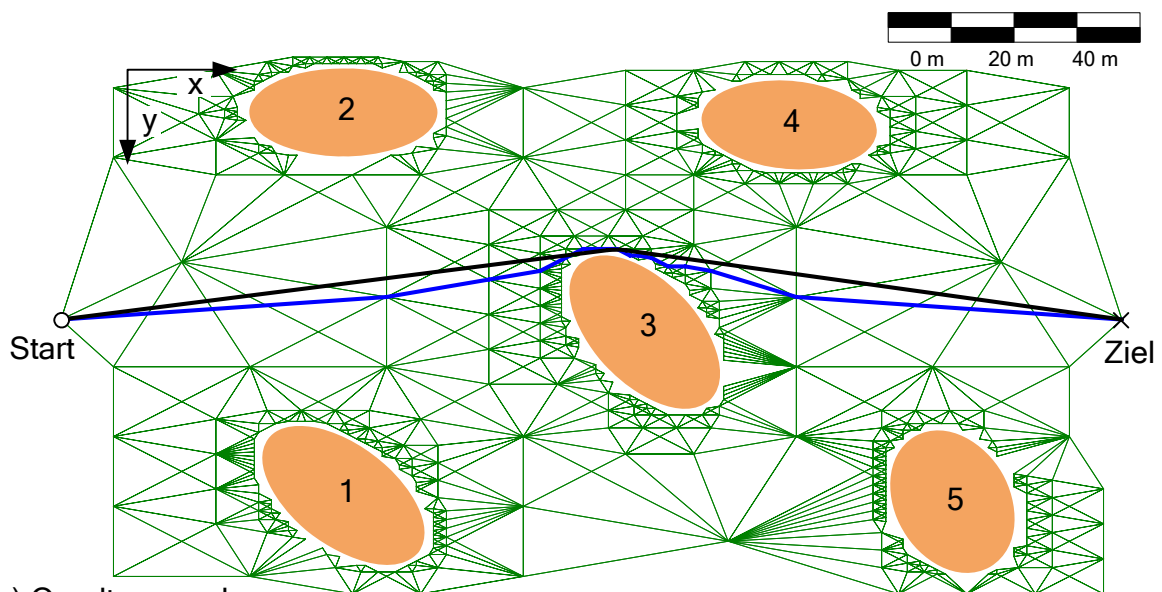
## A.3 Darstellung der geometrischen Graphen



a) Sichtbarkeitsgraph



b) Sektorisierter Sichtbarkeitsgraph



c) Quadtreegraph

Abbildung A.3: Geometrische Graphen von Test 1 des Hindernisparcours

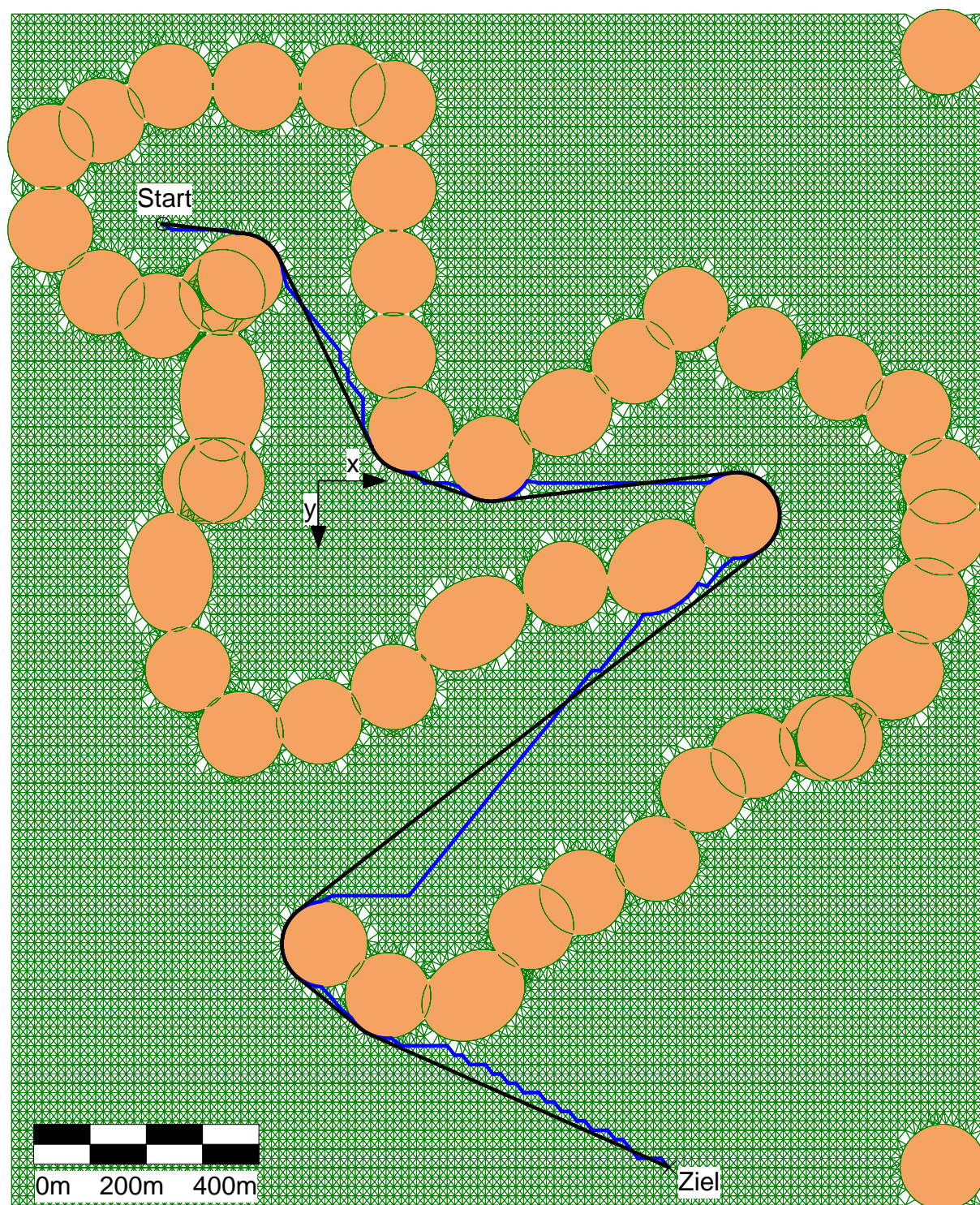


Abbildung A.4: Sektorisierter Sichtbarkeitsgraph des praktischen Testszenarios



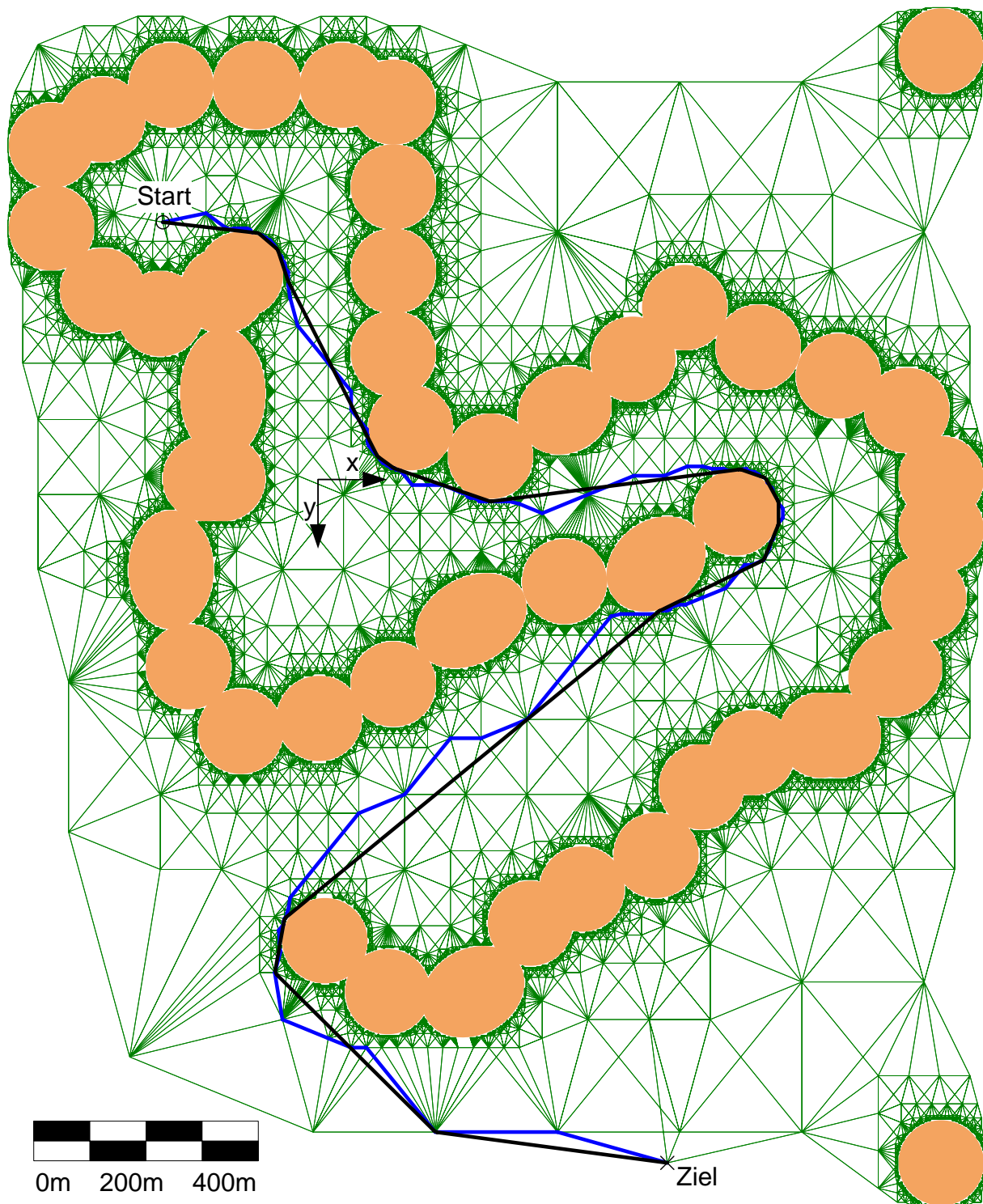


Abbildung A.5: Quadtreegraph des praktischen Testszenarios

# Literaturverzeichnis

- [1] ACKERMANN, JÜRGEN: *Robuste Regelung Analyse und Entwurf von linearen Regelungssystemen mit unsicheren physikalischen Parametern*. Berlin : Springer, 1993.
- [2] AHRNS, INGO: *Ultraschallbasierte Navigation und adaptive Hindernisvermeidung eines autonomen mobilen Roboters*. Kiel, Christian-Alberts-Universität, Institut für Informatik und Praktische Mathematik, Diplomarbeit, 1996.
- [3] ALEXANDRESCU, ANDREI: *Modernes C++ Design : Generische Programmierung und Entwurfsmuster angewendet*. Bonn : mitp-Verlag, 2003.
- [4] ALTENBURG, JENS: *Entwurf und Realisierung eines mobilen autonomen Systems variabler Intelligenz (MauSI®)*. Ilmenau, Technische Universität, Fakultät für Informatik und Automatisierung, Dissertation, 2004.
- [5] ALTENBURG, UWE: *Ein Beitrag zur Entwicklung von Strategien zur koordinierten Steuerung autonomer mobiler Multirobotersysteme*. Ilmenau, Technische Universität, Fakultät für Informatik und Automatisierung, Dissertation, 2004.
- [6] ARKIN, RONALD C.: *Behavior-based robotics*. Cambridge : MIT Press, 1998.
- [7] ASAKAWA, K.: Development of AUV: AQUA EXPLORER 1000. In: YUH, JUNKU (Hrsg.): *Underwater Robotic Vehicles: Design and Control*. Albuquerque; New Mexico USA : TSI Press, 1995.
- [8] ATLAS ELEKTRONIK GMBH: *DeepC WebSite*. 2005. – URL: <http://www.deepc-auv.de/deepc/DeepC.htm> [Stand 2005-01-08].
- [9] ATLAS ELEKTRONIK GMBH: *Mit freundlicher Genehmigung*. ATLAS ELEKTRONIK GMBH, 2006.
- [10] ATLAS ELEKTRONIK: *Software für Produkte Codier-Standards : Betriebsinternes Dokument*. 2001.
- [11] ATLAS MARIDAN APS: *MARIDAN WebSite*. 2006. – URL: <http://www.maridan.dk> [Stand 2006-02-18].
- [12] BALZERT, HEIDE: *Lehrbuch der Objektmodellierung : Analyse und Entwurf*. Heidelberg; Berlin : Spektrum Akademischer Verlag GmbH, 1999.
- [13] BARK, AXEL: *Sportküstenschifferschein + Sportbootführerschein See : mit Segelführerschein BR; mit amtlichen Fragekatalogen*. Bielefeld : Delius Klasing, 2001.
- [14] BARTEL, ROLAND: *Szeneninterpretation zur Erkennung von Konflikten an Engstellen*. Universität Stuttgart, Diplomarbeit, 1999.
- [15] BARTHEL, R. ; BESCHNIDT, J. ; GERN, T. GILLES, E.D.: Konfliktanalyse auf Binnenwasserstraßen. In: Schmidt, Günter ; Hanebeck, Uwe ; Freyberger, Franz (Hrsg.): *Autonome Mobile Systeme 1999 (15. Fachgespräch 26.-27. November 1999)*. Berlin : Springer-Verlag, 2000, S. 56-65.
- [16] BEEVERS, KRISTOPHER R. ; PENG, JUFENG: A\* graph search within the BGL framework. In: *Boost Graph Library 1.33.0* ([http://www.cs.rpi.edu/~beevek/research/astar\\_bgl04.pdf](http://www.cs.rpi.edu/~beevek/research/astar_bgl04.pdf) October 2004).
- [17] BERGEN, GINO VAN DEN: *Collision Detection in Interactive 3D Environments*. San Francisco : Morgan Kaufmann Publisher, 2004.
- [18] BGB II: Vertragsgesetz Seerechtsübereinkommen. (v. 02.09.1994) S. 1798 ff.
- [19] BÖHME, HANS-JOACHIM: *Serviceroboter und intuitive Mensch-Roboter-Interaktion*. Ilmenau, Technische Universität, Fakultät für Informatik und Automatisierung, Habilitationsschrift, 2001.
- [20] BOOST: *boost C++ Libraries*. 2006. – URL: <http://boost.org> [Stand 2006-02-25].
- [21] BORENSTEIN, J. ; KOREN, Y.: The Vector Field Histogram -Fast Obstacle Avoidance for Mobile Robots. In: *IEEE Journal of Robotics and Automation*, 7 (1991), Nr. 3, S. 278-288.
- [22] BORLAND GMBH: *Entwicklerhandbuch Band II : Borland C++ Builder 6 für Windows*. Langen : 2002.
- [23] BORLAND: *Borland WebSite*. 2006. – URL: <http://www.borland.com/> [Stand 2006-02-24].
- [24] BORRMANN, ALF ; LANDGRABE, GUNNAR ; MATÈRNE, JAN ; ET AL.: *Rational Rose und UML : Anleitung zum Praxiseinsatz*. Bonn : Galileo Computing GmbH, 2002.
- [25] BRAUN, BERNHARD ; CORSÉPIUS, RALF: AMOS: Schnelle Manipulator-Bewegungsplanung durch Integration potentialfeldbasierter lokaler und probabilistischer globaler Algorithmen. In: Schmidt, Günther ; Freyberger, Franz (Hrsg.): *Autonome Mobile Systeme 1996 (14. Fachgespräch München, 14.-15. Oktober 1996)*. Berlin : Springer-Verlag, 1996, S. 150-159.

- [26] BROCKHAUS IN TEXT UND BILD: *Ed. 2002 in der PC-Bibliothek (Version 3.0)*. Leipzig : Brockhaus AG, 2001.
- [27] BROCKHAUS, RUDOLF: *Flugregelung*. Berlin : Springer, 2001.
- [28] BRUCH, MICHAEL H. ; LUM, JASON ; YEE, SEE ;ET AL.: Advances in autonomy for small UGVs. In: *Proceedings of SPIE Defense and Security Symposium (SPIE Orlando, FL March 28 - April 1, 2005)*. 2005.
- [29] BRUTZMAN, DON ; HEALEY, TONY ; MARCO, DAVE; et al.: The Phoenix Autonomous Underwater Vehicle. In: KORTENKAMP, DAVID ; BONASSO, R. PETER ; MURPHY, ROBIN (Hrsg.): *Artificial Intelligence and Mobile Robots : Case Studies of Successful Robot Systems*. Menlo Park, CA : MIT Press, 1998, S. 323-360.
- [30] BRYSON, ARTUR E. ; HO, YU-CHI: *Applied optimal control : optimization, estimation and control* Ginn and Company, 1969.
- [31] BUCHBERGER, MARKUS: *Eine echtzeitfähige, reflexive Kollisionsvermeidung mit starker Bindung an eine Referenzbahn für einen Autonomen Mobilen Roboter*. Kaiserslautern, Universität, Fachbereich Informatik, Dissertation, 1996.
- [32] BUNDESMINISTERIUM FÜR UMWELT NATURSCHUTZ UND REAKTORSICHERHEIT (HRSG.): *Strategie der Bundesregierung zur Windenergienutzung auf See : im Rahmen der Nachhaltigkeitsstrategie der Bundesregierung*. Berlin : 2002.
- [33] CARTE MARINE OFFICIELLE: *Accès a la rade de brest 7401L*. Brest : Établissement Principal du Service Hydrographique et Océanographique de la Marine, 2004.
- [34] CGAL: *CGAL - Computational Geometry Algorithms Library*. 2006. – URL: <http://www.cgal.org/> [Stand 2006-07-02].
- [35] CHUHRAN, CHRISTOPHER D.: *Obstacle Avoidance Control for the REMUS Autonomous Underwater Vehicle*. Monterey, California, Naval Postgraduate School, Department of Mechanical Engineering, Master's Thesis, 2003.
- [36] COHEN, JONATHAN D. ; LIN, MING C. ; MANOCHA, DINESH ;ET AL.: I-COLLIDE: An Interactive and Exact Collision Detection System for Large-Scale Environments. In: *Proceedings of Symposium on Interactive 3D Graphics (I-COLLIDE: An Interactive and*. Monterey CA, USA : 1995,S. 189-218.
- [37] COX, NAGIN: The Mars Exploration Rovers: Hitting the Road on Mars. In: *Selected Plenaries, Milestones and Surveys of the 16th IFAC World Congress (16 IFAC Prague Czech Republic 4-8 July, 2005)*. 2005,S. 46-51.
- [38] CRIMMINS, DENISE ; DEACUTIS, CHRIS ; HINCHEY, ELIZABETH ;ET AL.: Use of a Long Endurance Solar Powered Autonomous Underwater Vehicle (SAUV II) to Measure Dissolved Oxygen Concentrations in Greenwich Bay, Rhode Island, U.S.A. In: *Conference Proceedings of IEEE Oceans'05 Europe (Oceans'05 Europe Brest, France June 20 -23, 2005)*. .
- [39] DEPARTMENT OF THE NAVY - UNITED STATES OF AMERICA: *The Navy Unmanned Undersea Vehicle (UUV) Master Plan*. 2004.
- [40] DICK, STEFAN ; KLEINE ECKARD ; MÜLLER-NAVARRA SYLVIN H.; ET AL.: *The Operational Circulation Model of BSH (BSHcmod) : Model description and validation*. Bundesamt für Seeschifffahrt und Hydrographie,2001.
- [41] DIJKSTRA, E.W.: A Note on Two Problems in Connexion with Graphs. In: *Numerische Mathematik*, (1959), Nr. 1.
- [42] EBERLY, DAVID H.: *3D Game Engine Design : A Practical Approach to Real-Time Computer Graphics*. San Diego : Academic Press, 2001.
- [43] EBERLY, DAVID H.: *e-Mail Korrespondenz: Minimal area ellipse*. [2004-07-12].
- [44] ECKSTEIN, JENS: *Echtzeitfähige Kollisionserkennung für Virtual Reality Anwendungen*. Saarbrücken, Universität des Saarlandes, Technische Fakultät, Dissertation, 1998.
- [45] EDLINGER, THOMAS: *Hierarchische Steuerung für einen mobilen Roboter zur autonomen Erkundung seiner Einsatzumgebung*. Fortschritt-Berichte VDI Reihe 8 Nr. 638. Düsseldorf : VDI Verlag, 1997.
- [46] EICHHORN, MIKE ; KUHN, THOMAS ; WERNSTEDT, JÜRGEN: Die Fuzzy Control Design Toolbox für MATLAB®. In: *at - Automatisierungstechnik*, 45 (1997), Nr. 11, S. 553-554.
- [47] EICHHORN, MIKE ; KUHN, THOMAS: *Fuzzy Control Design Toolbox: User Manuals*. München : SCIENTIFIC COMPUTERS, 1996.
- [48] EICHHORN, MIKE ; PFÜTZENREUTER, TORSTEN ; SCHMIDT, MANFRED ;ET AL.: Modellierung, Regelung und Navigation von Teilautonomen, Ferngesteuerten Unterwasserfahrzeugen. In: *at - Automatisierungstechnik*, 49 (2001), Nr. 8, S. 373-380.
- [49] EICHHORN, MIKE ; WERNSTEDT, JÜRGEN: Parameterermittlung mittels Parallelmodell - alte Konzepte mit neuen Möglichkeiten.: *Proceedings of 45th International Scientific Colloquium (45. IWK October 04-06, 2000)*. 2000.
- [50] EICHHORN, MIKE: *Gütekriterien und Suchverfahren für die Optimierung von Fuzzy-Systemen*. Technische Universität Ilmenau, Fakultät für Informatik und Automatisierung, Diplomarbeit, 1995.

- [51] FISCHER, KASPAR: *Fast and robust miniball of balls*. ETH Zürich, Institute for Theoretical Computer Science, 2003.
- [52] FISHMAN, GEORGE S.: *Monte Carlo : Concepts, Algorithms and Applications*. New York : Springer, 1995.
- [53] FLOHR, MANFRED: Software der Zukunft - Programmierter Absturz. In: *Chip*, September (2004), S. 138-145.
- [54] FLYGARE, HELENA: *Collision Avoidance for Autonomous Underwater Vehicles*. Lund, Institute of Technology, Department of Automatic Control, Master's thesis, 1995.
- [55] FÖRDERPROJEKT DES BMBF: DeepC - Aktiv autonomes Unterwasserfahrzeug für große Tauchtiefen : Förderkennzeichen: 03SX104E. Berlin : 2001.
- [56] FORESTI, GIAN LUCA: Detecting elliptical structures in underwater images. In: *Journal of Electronic Imaging*, 11 (2002), Nr. 1, S. 59-68.
- [57] FORSCHUNGSZENTRUM KARLSRUHE: *The KISMET 3D-Simulation Software*. 2006. – URL: <http://www-kismet.iai.fzk.de/> [Stand 2006-06-29].
- [58] FOSSEN, THOR I.: *Marine Control Systems : Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles*. Trondheim : Marine Cybernetics, 2002.
- [59] FOSSEN, THOR I.: Underwater Vehicle Dynamics. In: YUH, JUNKU *Underwater Robotic Vehicles: Design and Control*. Albuquerque, New Mexico USA : TSI Press, 1995, S. 15-40.
- [60] FREUND, E. ; KRÄMER, M. ; ROKOSSA, D.: Steuerung und Simulation autonomer, kooperativer Systeme. In: Wörn, Heinz ; Dillmann, Rüdiger ; Henrich, Dominik (Hrsg.): *Autonome Mobile Systeme 1998 (14. Fachgespräch Karlsruhe, 30.November-1.Dezember 1998)*. Berlin : Springer-Verlag, 1999, S. 198-207.
- [61] FURTWÄNGLER, RALF ; HANEBECK, UWE D. ; SCHMIDT, GÜNTHER: Dynamische Bänder zur Bewegungsplanung für mobile Manipulatoren. In: Wörn, Heinz ; Dillmann, Rüdiger ; Henrich, Dominik (Hrsg.): *Autonome Mobile Systeme 1998 (14. Fachgespräch 30.November-1.Dezember 1998)*. Berlin : Springer-Verlag, 1999, S. 164-171.
- [62] GÄRTNER, BERND ; SCHÖNHERR, SVEN: Smallest Enclosing Ellipses : An Exact and Generic Implementation in C++. 1998.
- [63] GESELLSCHAFT FÜR INERTIALE MESS-, AUTOMATISIERUNGS- UND REGELSYSTEME: *iMAR WebSite*. 2006. – URL: <http://www.imar-navigation.de/>.
- [64] GESELLSCHAFT FÜR MARITIME TECHNIK: Workshop zur "Strategischen Positionierung deutscher Unternehmen in der Offshoretechnik Öl und Gas". (23.November 2004). Hamburg : 2004.
- [65] GILBERT, ELMER G. ; JOHNSON, DANIEL W. ; KEERTHI, S. SATHIYA: A Fast Procedure for Computing the Distance Between Complex Objects in Three-Dimensional Space. In: *IEEE Journal of Robotics and Automation*, 4(2), 1988.
- [66] GOTTSCHALK, STEFAN ; LIN, MING ; MANOCHA, DINESH: OBB-Tree: A Hierarchical Structure for Rapid Interference Detection. In: *Computer Graphics (ACM SIGGRAPH '96 Proceedings August)*. 1996, S. 171-180.
- [67] GRACANIN, DENIS ; VALAVANIS, KIMON ; TSOURVELOUDIS, NIKOS C. ; ET AL.: Virtual-Environment-Based Navigation and Control of Underwater Vehicles. In: *IEEE Robotics & Automation Magazine*, June (1999), S. 53-62.
- [68] GREGOR, RUDOLF: *Fähigkeiten zur Missionsdurchführung und Landmarkennavigation*. München, Universität der Bundeswehr, Fakultät für Luft- und Raumfahrttechnik, Dissertation, 2002.
- [69] GULDNER, JÜRGEN: Autonomes Navigationssystem für einen mobilen Roboter oder Manipulator. Patentnummer: DE 4408982 C1, Anmeldung 16. März 1994.
- [70] GULDNER, JÜRGEN: Autonomous Navigation System for a Mobile Robot or Manipulator. Patentnummer: US 5758298, Anmeldung March 15, 1995.
- [71] GULDNER, JÜRGEN: *Intelligentes hierarchisches Regelungskonzept für autonome mobile Robotersysteme : Globale Planung - Lokale Kollisionsvermeidung - Robuste Bewegungsplanung*. Fortschritt-Berichte VDI Reihe 8 Nr. 509. Düsseldorf : VDI Verlag, 1995.
- [72] GULDNER, JÜRGEN: Lokale Kollisionsvermeidung für mobile Roboter mittels künstlicher harmonischer Dipol-Potentiale. In: *at - Automatisierungstechnik*, 45 (1997), Nr. 1, S. 24-35.
- [73] GUTMANN, JENS-STEFFEN ; NEBEL, BERNHARD: Navigation mobiler Roboter mit Laserscans. In: Levi, Paul ; Bräunl, Thomas ; Oswald, Norbert (Hrsg.): *Autonome Mobile Systeme 1997 (13. Fachgespräch Stuttgart, 6.-7. Oktober 1997)*. Berlin : Springer-Verlag, 1997, S. 36-47.
- [74] HAGLER, MATTHIAS: *Reaktive Hindernisvermeidung für einen mobilen Roboter*. Graz, Technische Universität Graz, Institut für Regelungstechnik, Diplomarbeit, 2004.
- [75] HELD, M. ; J. KLOSOWSKI, AND J. MITCHELL.: Evaluation of Collision Detection Methods for Virtual Reality Fly-Throughs. In: *In the Proceedings of the Seventh Canadian Conference on Computational Geometry (Evaluation of Collision Detect. August 10-13, 1995)*. Québec, Canada : 1995, S. 205-210.
- [76] HENRICH, DOMINIK ; GONTERMANN, STEFAN ; WÖRN, HEINZ: Schnelle Kollisionserkennung durch parallele Abstandsberechnung. In: Levi, Paul ; Bräunl, Thomas ; Oswald, Norbert (Hrsg.): *Autonome Mobile Systeme 1997 (13. Fachgespräch 6.Oktober-7.Oktober 1997)*. Springer-Verlag, 1997, S. 131-142.

- [77] HENRICH, DOMINIK ; GRAF, RENÉ: Flächendeckende Bahnplanung in vollständig, teilweise oder nicht bekannten Umgebungen. In: Wörn, Heinz ; Dillmann, Rüdiger ; Henrich, Dominik (Hrsg.): *Autonome Mobile Systeme 1998 (14. Fachgespräch 30.November-1.Dezember 1998)*. Berlin : Springer-Verlag, 1999, S. 188-197.
- [78] HOFNER, CHRISTIAN ; SCHMIDT, GÜNTHER: Path Planning and Guidance Techniques for an Autonomous Mobile Cleaning Robot. In: Graefe, Volker (Hrsg.): *Seventh International Conference on Intelligent Robots and Systems - IROS '94 (IROS '94 12-16 September 1994)*. Amsterdam : Elsevier Science B.V., 1995, S. 241-257.
- [79] HÖLSCHER-HÖBING, U. ; KALWA, JÖRG: *Konzept Objekterkennung : Interner Bericht DR4001G022BSG*. Bremen : ATLAS ELEKTRONIK, 2001.
- [80] HOPPEN, PETER: *Autonome Mobile Roboter : Echtzeitnavigation in bekannter und unbekannter Umgebung*. Mannheim : BI Wissenschaftsverlag, 1992.
- [81] HYLAND, JOHN C. ; TAYLOR, FRED J.: Mine Avoidance Techniques for Underwater Vehicles. In: *IEEE Journal of Oceans Engineering*, 18 (1993), Nr. 3, S. 340-350.
- [82] IBM: *Rational WebSite*. – URL: <http://www-306.ibm.com/software/rational> [Stand 2006-03-04].
- [83] IGS: *Sicherheitsroboter*. 2006. – URL: <http://www.igs-hagen.de/mosro.htm> [Stand 2006-07-17].
- [84] iMAR: *iMAR WebSite*. 2006. – URL: <http://www.imar-navigation.de> [Stand 2006-08-04].
- [85] ISERMANN, ROLF: *Digitale Regelungssysteme Band 1: Grundlagen Deterministische Regelungen*. Berlin : Springer, 1988.
- [86] JACOBI, MARCO: *Untersuchung von Bahnplanungsalgorithmen für das Autonome Unterwasserfahrzeug "DeepC"*. Ilmenau, Technische Universität Ilmenau, Fakultät für Informatik und Automatisierung, Studienjahresarbeit, 2004.
- [87] JAMES, DOUG L. ; PAI, DINESH K.: BD-Tree: Output-Sensitive Collision Detection for Reduced Deformable Models. In: *ACM Transactions on Graphics (SIGGRAPH 2004)*, 23 (2004), Nr. 3, S. 393-398.
- [88] KÄMPKE, THOMAS ; STROBEL, MATTHIAS: Navigation nicht-kreisförmiger mobiler Roboter in hindernisdichten Umgebungen. In: Wörn, Heinz ; Dillmann, Rüdiger ; Henrich, Dominik (Hrsg.): *Autonome Mobile Systeme 1998 (14. Fachgespräch 30.November-1.Dezember 1998)*. Berlin : Springer-Verlag, 1999, S. 156-163.
- [89] KARIMANZIRA, DIVAS ; OTTO, PETER ; WERNSTEDT, JÜRGEN: A Fuzzy-Based manoeuvre Management System for an Autonomous Underwater Vehicle. In: *Preprints of the 16th IFAC World Congress (16th IFAC World Congress July 3-8, 2005)*. Prague : 2005.
- [90] KHATIB, OUSSAMA: Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. In: *The International Journal of Robotics Research*, 5 (1986), Nr. 1, S. 90-98.
- [91] KLEIN, ROLF: *Algorithmische Geometrie : Grundlagen, Methoden, Anwendungen*. Berlin : Springer, 2005.
- [92] KNIERIEMEN, THOMAS: *Autonome Mobile Roboter : Sensordateninterpretation und Weltmodellierung zur Navigation in unbekannter Umgebung*. Mannheim : BI Wissenschaftsverlag, 1991.
- [93] KOCH, MARIO ; KUHN, THOMAS ; WERNSTEDT, JÜRGEN: *Fuzzy Control*. München : R. Oldenbourg Verlag GmbH, 1996.
- [94] KONDO, HAYATO ; MAKI, TOSHIHIRO ; URA, TAMAKI ; ET AL.: Observation of Breakwaters and their Rock Mound by AUV "Tri-Dog 1" at Kamaishi Bay. In: *Proceedings of OCEANS'05 Europe IEEE Conference & Exhibition (IEEE OCEANS '05 Europe Brest June 20-23, 2005)*. 2005.
- [95] KONDO, HAYATO ; MAKI, TOSHIHIRO ; URA, TAMAKI ; ET AL.: Structure Tracing with a Ranging System using a Sheet Laser Beam. In: *Proceedings of 2004 International Symposium on Underwater Technology (UT 2004 Taipei, Taiwan 20.-23. April 2004)*. 2004, S. 83-88.
- [96] KONGSBERG MARITIME: Six Years of HUGIN AUV Offshore Surveying. In: *Business Briefing: Exploration & Production: The Oil & Gas Review*, (2004),.
- [97] KOREN, Y. ; BORENSTEIN, J.: Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation. In: *Proceedings of the IEEE Conference on Robotics and Automation (IEEE Conference on Robotics and Automation April 7-12 1991)*. Sacramento : 1991, S. 1398-1404.
- [98] KRUSE, ECKHARD: *Bewegungsplanung für mobile Roboter in dynamischen Umgebungen auf Basis automatisch erzeugter Daten*. Aachen : Shaker Verlag, 1998.
- [99] KUHLINS, STEFAN ; SCHADER, MARTIN: *Die C++ Standardbibliothek : Einführung und Nachschlagewerk*. Berlin; Heidelberg; New York; Barcelona; Hongkong; London; Mailand; Tokio : Springer, 2002.
- [100] KUHN, THOMAS: Adaption von Flugreglern mittels Fuzzy-Logik. In: *at-Automatisierungstechnik*, 48 (2000), Nr. 12, S. 597-601.
- [101] KÜMPEL, JÖRG: *Spezielle Zellerteilungsverfahren zur globalen Bahnplanung autonomer mobiler Roboter*. Fortschritt-Berichte VDI Reihe 8 Nr. 955. Düsseldorf : VDI Verlag, 2002.

- [102] LANE, D.M., TRUCCO, E.: Embedded Sonar & Video Processing for AUV Applications. In: *Offshore Technology Conference (Offshore Technology Conference 1-4 May 2000)*. Houston, Texas : 2000.
- [103] LIEBEZEIT, THOMAS: *Missionsbezogener modellgestützter Entwurf mobiler autonomer Systeme*. Ilmenau, Technische Universität, Fakultät für Informatik und Automatisierung, Dissertation, 2005.
- [104] MAGNENAT-THALMANN, NADIA ; THALMANN, DANIEL: *Handbook of Virtual Humans*. Chichester, West Sussex : John Wiley & Sons, Ltd, 2004.
- [105] MARTINS, ALFREDO ; ALMEIDA, JOSÉ MIGUEL ; SILVA, EDUARDO ; ET AL.: Hybrid Maneuver for Gradient Search with Multiple Coordinated AUVs. In: *Proceedings in 5t IFAC Symposium on Intelligent Autonomous Vehicles (IAV 2004 Lisboa July 2004)*. 2004.
- [106] MATHAI, A. M.: *An Introduction to Geometrical Probability*. Amsterdam : Gordon and Breach Science Publishers, 1999.
- [107] MCLEAN, CRAIG: Ocean Exploration: Building Innovative Partnerships in the Spirit of Discovery. In: *Proceedings of 2004 International Symposium on Underwater Technology (UT 2004, Taipei, Taiwan 20.-23. April 2004)*. 2004, S. 3-6.
- [108] MEZGER, JOHANNES: *Effiziente Kollisionsdetektion in der Simulation von Textilien*. Tübingen, Universität Tübingen, Wilhelm-Schickard-Institut für Informatik, Diplomarbeit, 2001.
- [109] MICROSOFT: *About Timers*. 2006. – URL: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/winui/winui/windowsuserinterface/windowing/timers/abouttimers.asp> [Stand 2006-03-30].
- [110] MINGUEZ, JAVIER ; MONTANO, LUIS: Nearness Diagram (ND) Navigation: Collision Avoidance in Troublesome Scenarios. In: *IEEE Transactions on Robotics and Automation*, 20 (2004), Nr. 1, S. 45-59.
- [111] MINGUEZ, JAVIER ; MONTANO, LUIS: Nearness Diagram Navigation (ND): A New Real Time Collision Avoidance Approach. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000 31 Oct. - 5 Nov. 2000)*. Takamatsu : 2000, S. 2094-2100.
- [112] MIRTICH, BRIAN: Efficient Algorithms for Two-Phase Collision Detection. In: GUPTA, KAMAL ; POBIL, ANGEL PASQUAL DEL. (Hrsg.): *Practical Motion Planning in Robotics*. Chichester : John Wiley & Sons Ltd, 1998, S. 203-223.
- [113] MOJAEV, ALEXANDER ; ZELL, ANDREAS: Aufbau topologischer Karten und schnelle globale Bahnplanung für mobile Roboter. In: Schmidt, Günther ; Hanebeck, Uwe ; Freyberger, Franz (Hrsg.): *Autonome Mobile Systeme 2001 (17. Fachgespräch Stuttgart, 11.-12. Oktober 2001)*. Berlin : Springer-Verlag, 2001, S. 164-170.
- [114] MOJAEV, ALEXANDER ; ZELL, ANDREAS: Robuste reaktive Bahnregelung und Kollisionsvermeidung eines autonomen mobilen Roboters. In: Schmidt, Günther ; Hanebeck, Uwe ; Freyberger, Franz (Hrsg.): *(15. Fachgespräch, München 26.-27. November 1999)*. Berlin : Springer-Verlag, 2000, S. 284-292.
- [115] MORAVEC, HANS P.: *Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rove*, Technical Report CMU-RI-TR3, Pittsburgh, PA: Carnegie-Mellon University, Robotics Institute, 1980.
- [116] NAEEM, W. ; SUTTON, R. ; AHMAD, S.M.: LQG/LTR Control of an Autonomous Underwater Vehicle : Using a Hybrid Guidance Law. In: *Guidance and Control of Underwater Vehicles 2003 (GCUV 2003 IFAC Workshop Newport, South Wales 9-11 April 2003)*. Amsterdam : Elsevier LTD, 2003, S. 31-36.
- [117] NILSSON, NILS.J: *Principles of Artificial Intelligence*. Berlin : Springer, 1982.
- [118] Norm E DIN 13312 *Navigation : Begriffe, Abkürzungen, Formelzeichen, graphische Symbole*.
- [119] OTTMANN, THOMAS ; WIDMAYER, PETER: *Algorithmen und Datenstrukturen*. Heidelberg : Spektrum Akademischer Verlag, 2002.
- [120] PAPAGEORGIOU, MARKOS: *Optimierung : Statische, dynamische, stochastische Verfahren für die Anwendung*. München : Oldenbourg, 1991.
- [121] PÉREZ, MARC CARRERAS I: *An overview of Behavioural-based Robotics with simulated implementations on an Underwater Vehicle*. Girona, Universitat de Girona, Institut d'Informàtica i Aplicacions, Report, 2000.
- [122] PETILLOT, YVAN ; RUIZ, IOSEBA TENA ; LANE, DAVID M.: Underwater Vehicle Obstacle Avoidance and Path Planning Using a Multi-Beam Forward Looking Sonar. In: *IEEE Journal of Oceans Engineering*, 26 (2001), Nr. 2, S. 240-251.
- [123] PETERSSON, PER OLAF: *Sampling-based Path Planning for an Autonomous Helicopter*. Linköping, Sweden, Linköping universitet, Department of Computer and Information Science, Thesis, 2006.
- [124] PETERSSON, RER OLAF: *Helicopter Path Planning using Probabilistic Radmaps*. Linköping, Sweden, Linköping universitet, Department of Computer and Information Science, Master's thesis, 2003.
- [125] PFAFF, GERHARD ; MEIER, CHRISTOF: *Regelung elektrischer Antriebe II - Geregelte Gleichstromantriebe*. München : Oldenbourg, 1988.
- [126] RAUSCH, WILLI ALEXANDER: *Interaktion autonomer Roboter : Kooperative Navigation autonomer Roboterfahrzeuge unter Einsatz von Funk und Ultraschall*. Aachen : Shaker, 1998.
- [127] REINERT, DIETMAR ; SCHAEFER, MICHAEL ; BÖMER, THOMAS : Regeln für den Entwurf und die Programmierung sicherheitsbezogener Software. In: *atp Automatisierungstechnische Praxis*, 41 (1999), Nr. 6, S. 21-30.



- [128] REINISCH, KARL: *Analyse und Synthese kontinuierlicher Regelungs- und Steuerungssysteme*. Berlin : Verlag Technik GmbH, 1996.
- [129] SALONIA, MARCOS: *Mit freundlicher Genehmigung*. Universität Karlsruhe, Institut für Prozessrechenstechnik, Automation und Robotik (IPR), 2006.
- [130] SAYYAADI, HASSAN ; URA, TAMAKI: AUVs Dynamics Modeling, Position Control and Path Planning using Neural Networks. In: *Proceedings of the IFAC conference on control applications in marine systems (CAMS'2001 July 17-20 2001)*. Glasgow, Scotland, UK : 2001.
- [131] SBS: *SBS Technologies WebSite*. 2006. – URL: <http://www.sbs.com>.
- [132] SCHEYER, WERNER: *Orientierungstauchen - Nachttauchen - Strömungstauchen - Wracktauchen*. Bielefeld : Delius Klasing, 2002.
- [133] SCHLETTE, CHRISTIAN: *Bahnplanung für autonome Systeme*. 2000. – URL: <http://www.irf.de/seminar/schlette/welcome.htm> [Stand 2004-12-22].
- [134] SCHMIDT, PETER: *Hard- und softwaretechnische Realisierung einer Steuerung für einen mobilen Roboter mittels Fuzzy-Logik*. Ilmenau, Technische Universität Ilmenau, Fakultät für Informatik und Automatisierung, Diplomarbeit, 1995.
- [135] SCHNEIDER, PHILIP J. ; EBERLY, DAVID H.: *Geometric Tools for 3D Graphics*. San Francisco : Morgan Kaufmann Publishers, 2003.
- [136] SCHÖNHERR, SVEN: *e-Mail Korrespondenz: Smallest enclosing Ellipses*. [2004-07-18].
- [137] SIEBEL, NILS ; LANG, OLIVER ; WIRTH FABIAN, GRÄSER, AXEL: Robuste Positionierung eines Roboters mittels Visual Servoing unter Verwendung einer Trust-Region-Methode. In: *Forschungsbericht Nr. 99-1 der Deutschen Forschungsvereinigung für Meß-, Regelungs- und Systemtechnik (DFMRS) e.V. (DFMRS Jahrestagung 1999 zum Thema Autonome Prozessführung, Prozessüberwachung und Montage, Bremen 11.-12. November 1999)*. 1999, S. 23-39.
- [138] SIEK, JEREMY, G. ; LEE, LIE-QUAN ; LUMSDAINE, ANDREW: *The Boost Graph Library : User Guide and Reference Manual*. New York : Addison-Wesley, 2002.
- [139] SPANGELO, INGE ; EGELAND, OLAV: Trajectory Planning and Collision Avoidance for Underwater Vehicles Using Optimal Control. In: *IEEE Journal of Oceanic Engineering*, 18 (1994), Nr. 4, S. 502-511.
- [140] TELEDYNE RDI: *DVLnav WebSite*. – URL: <http://www.dvlnav.com> [Stand 2006-03-05].
- [141] THE MATHWORKS: *Nonlinear Control Design Blockset User's Guide*. Natick USA : The MathWorks, Inc., 2002.
- [142] THE MATHWORKS: *The MathWorks Homepage*. 2006. – URL: <http://www.mathworks.com> [Stand 2006-06-29].
- [143] TRITECH: *Tritech International WebSite*. 2006. – URL: <http://www.tritech.co.uk> [Stand 2006-02-25].
- [144] TU ILMENAU: *Abschlussbericht BMBF-Verbundvorhaben DeepC*. 2006. – URL: <http://www.tu-ilmenau.de/site/systemanalyse/fileadmin/template/startIA/sanalyse/div/projekte/DeepC-Internet.pdf> [Stand 2006-06-29].
- [145] V. HINÜBER, EDGAR: *e-Mail Korrespondenz: Positioniergenauigkeit INS 2006*.
- [146] V. HINÜBER, EDGAR: *New Approaches in High-Performance Navigation Solutions for AUVs and ROVs*. 2006. – URL: [http://www.imar-navigation.de/download/underwater\\_imar.pdf](http://www.imar-navigation.de/download/underwater_imar.pdf).
- [147] VERBEEK, CHRISTIAN: *Reaktive Steuerung autonomer mobiler Roboter*. Bielefeld, Universität, Technische Fakultät, Dissertation, 2001.
- [148] VERHOEF, JACOB: Canada's Claim for an Extended Continental Shelf: Requirements for Data Collection under the Ice, Role of AUV's. In: *Proceedings of the 4th Biannual NRC-IOT Workshop (4th Biannual NRC-IOT Workshop on Underwater Vehicle Technology: Vehicles for Long -Endurance Missions St. John's Newfoundland Kanada 14.-15. Oktober 2004)*. St. John's : 2004.
- [149] VERTH, JAMES M. VAN ; BISHOP, LARS M.: *Essential Mathematics for Games and Interactive Applications*. San Francisco : 2004.
- [150] WANG, YONGJI ; LANE, DAVID M. ; FALCONER, GAVIN J.: Two novel approaches for unmanned underwater vehicle path planning: constrained optimisation and semi-infinite constrained optimisation. In: *Robotica*, 18 (2000), Nr. 2, S. 123-142.
- [151] WARKEN, THOMAS: *Collision Detection for Curved Rigid Objects in the Context of Dynamics Simulations*. Saarbrücken, Universität des Saarlandes, Naturwissenschaftlich-technische Fakultät, Dissertation, 2004.
- [152] WATANABE, K. ; HASHEM, M.M.A. ; IZUMI, K.: Global Path Planning of Mobile Robots as an Evolutionary Control Problem. In: *Proceedings of European Control Conference (ECC'99 Karlsruhe 31. August-3. September)*. 1999.
- [153] WEISSTEIN, ERIC W.: *CRC Concise Encyclopedia of Mathematics*. Boca Raton, Florida : Chapman & Hall/CRC, 2003.
- [154] WERNLI, ROBERT L.: AUV Commercialization-Who's Leading the Pack? In: *Proceedings of MTS/IEEE OCEANS 2000 ( Providence, Rhode Island, U.S.A September 11-14September 11-14, 2000)*. 2000.

- [155] WIKIPEDIA: *Wikipedia Die freie Enzyklopädie*. 2006. – URL: <http://de.wikipedia.org/wiki/Hauptseite> [Stand 2006-02-16].
- [156] XERCES: *Xerces WebSite*. 2006. – URL: <http://xerces.apache.org> [Stand 2006-02-25].
- [157] ZHANG, JIANWEI: *Ein integriertes Verfahren zur effizienten Planung und Ausführung von Roboterbewegungen in unscharfen Umgebungen*. Sankt Augustin : Infix, 1995.